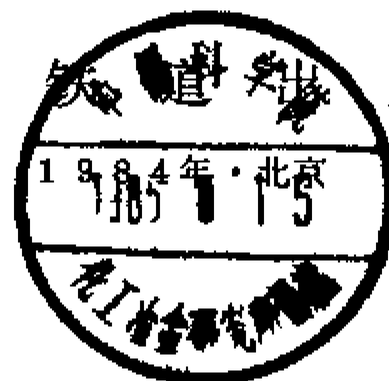
图论基础及应用

北方交通大学 吴文虎 主编



中国铁道出版社

1984年·北京

内 容 简 介

本书系统地介绍了图论的基本理论和在若干工程技术方面的应用。

全书共十图章，前七章为图论基础部分，后七章为图论应用部分。其中，第十一章图论算法和计算机程序可作为计算工具使用。应用部分偏重于电网络、开关网络和交通网络这三方面。每章后附有若干习题，以资练习。本书适用于铁路行车自动控制、通信、计算机科学和运输各专业，可以作为铁路高等院校和一般高等工科院校相应专业的研究生课或大学高年级选修课的教材或参考书。对于图论有兴趣的工程技术人员亦可作为自学参考书。

图论基础及应用

北方交通大学 吴文[illegible] 主编

中国铁道出版社出版

新华书店北京发行所发行

各 地 新 华 书 店 经 售

中国铁道出版社印刷厂印

开本：787×1092 1/16 印张 19.5 字数：486千

1984年9月 第1版 1984年9月 第1次印刷

印数：0001—7,500册 定价：3.65元

序

图论是一门研究事物之间相互关系或联系的学科，它用一组点来代表事物，用一组边来代表不同事物之间的关系，形成一个抽象图形来研究点与边之间的特性。这门学科的开创要追溯到1736年，到现在已有二百多年历史。近二、三十年来，尤其在高速电子计算机问世后，它有较快的发展。这是因为电子计算机的发展和应用，为研究庞大复杂的系统提供了可能性。现在图论的应用范围很广，不但能应用于自然科学，也能应用于社会科学。

为了使图论这一理论能够应用到铁路企业各方面的研究中，编者于1978～1979年编写了一本《图论基础及应用》作为讲义，在北方交通大学进行讲授。讲义内容主要取材于本书参考文献中第一、第二两本书。讲义共分十五章，前七章为图论基础部分，后八章为图论应用部分。应用部分偏重于电网络、开关网络和交通网络这三方面。

通过几次讲课，编者根据同学们的反映和积累的新资料，对原来的讲义作了部分增删。现在全书共分十四章，前七章仍为图论基础部分，后七章为图论应用部分，而第十一章图论算法和计算机程序作为应用时的共同必需工具。

本书编写的目的是为了适应铁路行车自动控制、通信、计算机科学和运输各专业和一般高等工科院校相应专业的研究生作教材或参考书用。每章后附有若干习题，以资练习。全书内容较多，各专业可以根据需要和研究方向删除某些章节，或增加某些适合研究方向的内容，开设一门五、六十学时的研究生课。本书前七章图论基础部分可以作为大学高年级学生的选修课。

本书除第八、九两章由许珠副教授改写外，其余部分均由吴文泷教授编写。第十四章中“接点开关电路中迂回电路的诊断”的部分内容由汪焕敏讲师提供素材。

本书承蒙中国科学技术大学研究生院的左垲教授审阅，提出了许多宝贵意见，在此谨致衷心的感谢。

由于编者的水平有限，书中可能存在若干错误和缺点，请读者不吝批评指正。

编　者

1983年3月于北方交通大学电信系

目　　录

第一章 绪　论

第一节 图论的发展简史

在普鲁士的哥尼斯堡有一条河，河中有两个岛A和B，为了岛A和B与两旁的陆地C和D相连通，岛与陆地、岛与岛之间建有a，b，c，d，e，f和g七座桥，如图1－1所示。

有人曾提出了这样一个问题，一个人能否一次走遍两岛、两旁陆地和7座桥，而每座桥又只能经过一次呢？1736年瑞士数学家欧拉回答了这个问题。他提出了一篇论文，讨论了哥尼斯堡桥的问题，文章的结论是一个人不可能一次走遍两岛，两旁陆地和7座桥，而每座桥又只能经过一次。这是图论方面已知的最早的文章，人们认为图论由此发端。这个问题其实就是能否每条边不重复不遗漏地一笔画出一个图形的问题。至于为什么不可能呢？我们将在第二章中讲到。

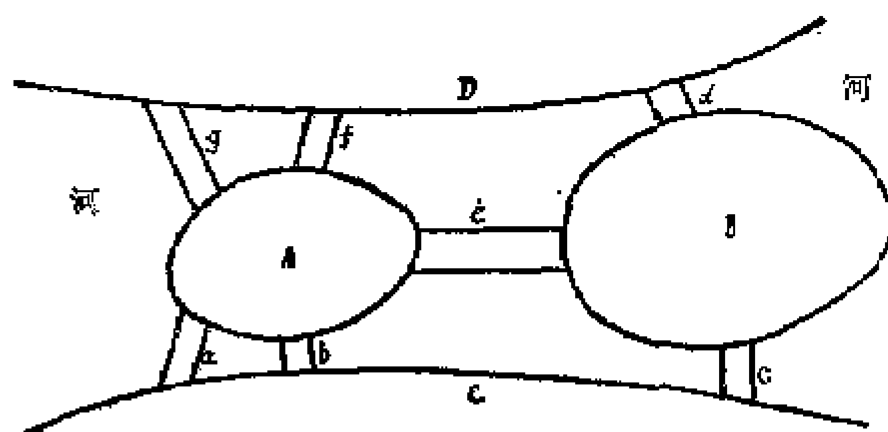


图 1－1

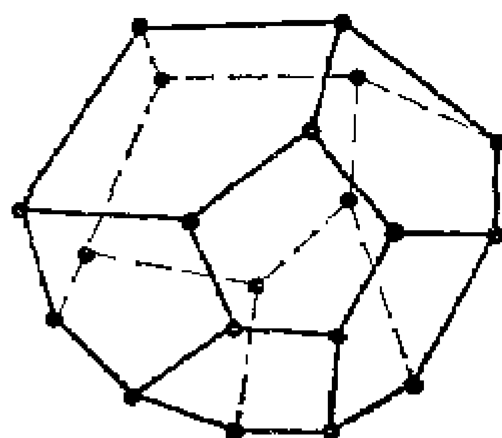


图 1－2

1847年克希荷夫应用图论的方法来分析电网络，奠定了现代网络理论的基础，这就是电工原理中的克希荷夫电流定律和克希荷夫电压定律。在1857年，凯莱在试算饱和碳化氢的同分异性体时，发现了“树”的概念。在这段时间里，对图论的发展另有两个里程碑。一个是莫别斯约于1840年提出一个“四色推测”，这个问题是说一幅地图只要用4种不同颜色着色，就可以使互相接壤的国家由不同的颜色来区分。（这个问题当时称为推测，一直到1976年才由美国一大学用电子计算机加以证明）。另一个是汉密尔顿于1859年发明了一个小玩具，这个玩具是一个木刻的正12面体，每面系正五角形，三面相交于一角，共有20个角，如图1－2所示。每角标有世界上一个重要城市。玩具提出一个谜，要求沿正12面体的边寻找一条路通过20个城市，而每个城市只通过一次，最后返回原地。此后约有半个世纪研究的人不多。直到1936年哥尼格发表了第一本图论专著，从此图论成为一门独立的学科。最近二三十年来，图论有了较大的发展，人们对线性图的理论研究和各方面的广泛应用进行了大量工作。

当应用图论来解决实际问题时，不管是电网络的分析，电路设计，数据的结构，或社会

科学方面的问题，几乎需要引出复杂的图形，这些图形，实际上如果没有电子计算机的帮助是不大可能分析的。近年来，大家所以对于图论研究的兴趣又高涨起来，高速数字计算机的出现是原因之一。

图论应用的范围很广，它不但能应用于自然科学，也能应用于社会科学。它非但广泛应用于电信网络、电力网络、运输能力、开关理论、编码理论、控制论、反馈理论、随机过程、可靠性理论、化学化合物的辨认、计算机的程序设计、故障诊断、人工智能、印刷电路板的设计、图案识辨、地图的着色、情报检索，也应用于诸如语言学、社会结构、经济学、运筹学、兵站学、遗传学等等方面。

图论应用范围虽然已经这样广泛，但是到目前为止，还有若干问题没有解决，相信通过对图论更深入的研究，图论必将得到更大的发展，它的应用范围将更为扩大。

第二节　集合论的一些基本概念

在日常生活和科学技术领域中，常常需要研究一类事物之间的相互关系，象球队和球队之间的竞赛关系；父母、兄弟或父子之间的亲属关系；电路的各个状态之间的相互转换关系等等。总之，研究一组事物之间的相互关系，我们抽象地用一组点来表示事物，用这些点间连接线表示事物与事物之间的相互关系。用这种方法研究一些具体问题的理论就是图论。

图论中常常把图看成一个抽象的代数系统，为此，图论也经常要用到代数学的一个重要基础——集合或组。我们在这里顺便介绍一些简单的集合论概念。图论常常用集合论中一些符号，为便利没有接触过集合论的读者，我们先介绍一些常用的符号。

集合是一种不可精确定义的代数学基本概念，一般只能给出直观性的描述。例如，甲工厂所属的全体车间是一个集合；称为甲厂车间集合；C地区全体工厂是一个集合，称为C地区工厂集合；整数的全体是一个集合，称为整数集合；而素数的全体称为素数集合。

集合中的每个成员称为元素，上面所说的那些集合中的车间、工厂、整数和素数都是集合中的元素。如果甲厂有a、b、c、d和e 5个车间，那么可以说元素a属于甲厂车间集合，记为

$$a \in \text{甲厂车间集合}$$

若另有一个f车间，它不是甲厂的一个车间，则f这个元素不属于甲厂车间集合，记为

$$f \notin \text{甲厂车间集合}$$

一般集合用大写英文字母，而元素则用小写英文字母。所以如果甲厂车间集合用S表示，则$a \in S$；$b \in S$；$c \in S$；$d \in S$；$e \in S$。这里需要说明一点，a、b、c、d、e这5个车间都是属于S这个集合的，而S是指甲厂中车间的集合，而不是指甲厂中其它别的什么。现在进一步来看看C地区，C地区有许多工厂，甲厂是其中的一个，假如这些工厂我们用S_1、S_2、S_3、S_4来表示，那么有

$S_1 \in C$；$S_2 \in C$；$S_3 \in C$；$S_4 \in C$这表明S_1、S_2、S_3和S_4 4个工厂都属于C地区的。

C地区车间的集合，是指由C地区各个工厂中的每一个车间为元素的集合。如果C地区这4个工厂中，S_1厂有5个车间；S_2厂有3个车间；S_3厂中有4个车间；S_4厂有7个车间，则C地区车间的集合就是指这19个车间。换言之，这19个车间都是C的一个元素，都属于C。若其中某个S_1厂含有a_1、b_1、c_1、d_1和e_1 5个车间，则元素a_1、b_1、c_1、d_1和e_1既属于S也属于C。于是定义S和C两个集合，如果S的每个元素都属于C集合的元素，则称S为C的

子集，即S包含于C中。根据这个定义，C本身也包含于C中，所以C也是C的子集，所以，子集S包含于C或等于C，因此记作

$$S\subseteq C$$

若S是C的子集，即$S\subseteq C$，而$S\neq C$，则称S为C的真子集或真子组，记为

$$S\subset C$$

显然，符号$\subset$意思指"包含"，而符号$\subseteq$有两重含义，既可"包含"，也可以"等于"。可见，真子集是指那些除集合本身以外的子集，真子集的元素必定比原集合的元素少。

上述集合C可以写成$C=\{(a, b, c, d, e), (f, g, h), (i, j, k, l), (m, n, o, p, q, r, s)\}$

由若干个集合组成的集合叫集体。C也是一个集体。

集合的表示方法一般有3种：

第一种称为描述法。就是在{ }内用一个句子来描述这个集合。例如：{甲厂全体车间}，{全体整数}。第二种称为列举法。列举法是将集合内的所有元素都列举在{ }内，并用逗号"，"隔开。例如：甲厂车间集合用S表示，甲厂的5个车间用a、b、c、d和e表示，则

$$S=\{a, b, c, d, e\} \tag{1-1}$$

又如整数集合用I表示

则

$$I=\{0, +1, -1, +2, -2, \cdots\cdots\} \tag{1-2}$$

第三种称为公式法

$$S=\{x \mid P(x)\} \tag{1-3}$$

式中 x是S的元素，这些元素满足$P(x)$条件，也就是说，若x_1代入$P(x)$，$P(x_1)$满足，或者说$P(x_1)$为真时，$x_1\in S$。

例如

$$S=\{0, 1, -1, 2, -2, 3, -3\}$$

用公式法可以表示为

$$S=\{x \mid x\in I\wedge |x|\leqslant 3\}$$

式中I是整数集合，符号$\wedge$表示"与"运算。

第三节 图的定义

图是用来研究一组具体事物之间相互关系的抽象代数。用点表示事物，用边表示事物与事物之间的关系，则图就可以用如图1—3的图形来表示。

线性图，或简称图的最普通的表示法。其中用黑点表示顶点，以后简称点，用线段表示边，所以线性图就是点组、边组以及点组和边组的关系的集体。它可以定义为

$$G=(V, E, \Psi) \tag{1-4}$$

式中 $V=\{1, 2, 3\cdots\cdots n_v\}$——点的有限非空组；

$E=\{a, b, c\cdots\cdots n_e\}$——边的有限组；

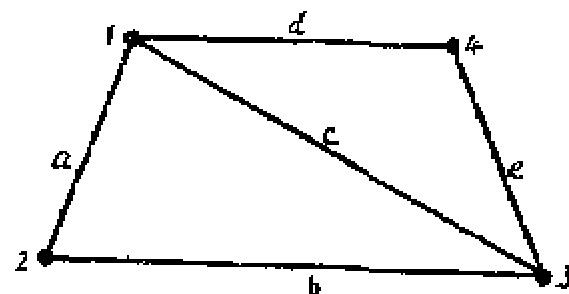


图 1—3

Ψ——V和E的对应关系，或称映射

向图。如果每条边上标有方向，如图 1 — 4 所示，这条边 a 称为有向边，有向边 a 的点 1 称为起点，点 2 称为终点。由有向边构成的图称为有向图，有向图中 E 和 V 的关系用 Ψ 表示，所以有向图的定义为

图 1 — 4

$$G=(V,\ E,\ \Psi) \qquad (1-5)$$

在无向图（如图 1 — 3）中，E 和 V 的关系 Ψ 为

$$a \to (1,\ 2)$$
$$b \to (2,\ 3)$$
$$c \to (1,\ 3)$$
$$d \to (1,\ 4)$$
$$e \to (3,\ 4)$$

符号→表示边的两个端点在右边的（ ）中，它们不分起点和终点，所以 a 边也可以写为

$$a \to (2,\ 1)$$

由于图只是 V，E，Ψ 的集体，因此每条边的形状是无关紧要的，所以 1 与 2 之间的边不论用图（1 — 5）中任何一种方式表示都是一样的，我们关心的只是几个点，几条边以及边和点之间如何联接，至于这些边的直和曲，长和短都是可以任意选定的。例如在图 1 — 6 中，同样性质的图可以画成几种不同的形式，只要保持点和边关系不变。

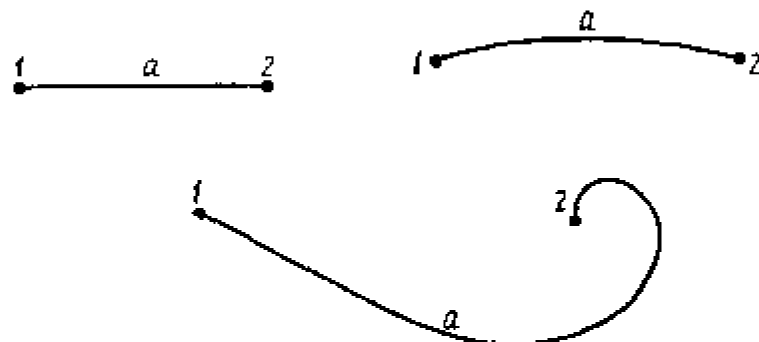
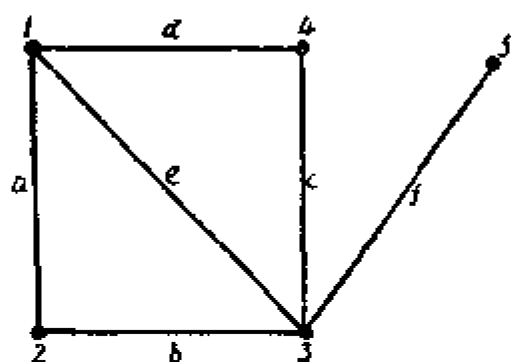


图 1 — 5

有向图的边是有向边，所以 E 和 V 的关系 Ψ 就不象无向图时两个端点可以任意调换，不分次序，图 1 — 7 就是一个有向图。



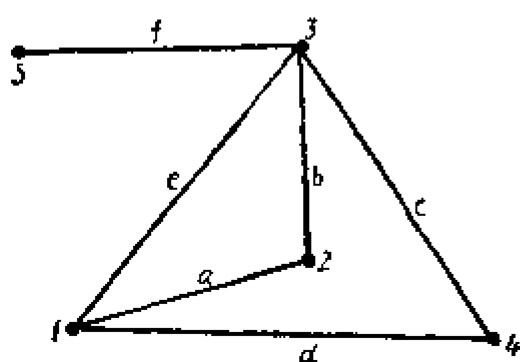


图 1 — 6

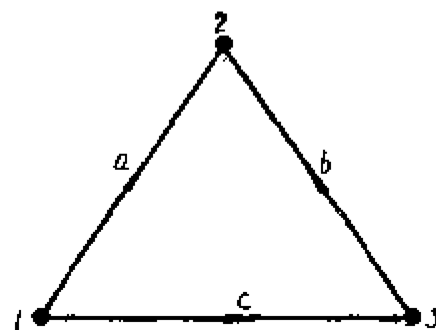


图 1 — 7

有向图 G 可以用下式表示

$$G=(V,\ E,\ \Psi)$$
$$V=\{1,\ 2,\ 3\}$$
$$E=\{a,\ b,\ c\}$$

Ψ 为

$$a \to (1,\ 2)$$
$$b \to (3,\ 2)$$
$$c \to (1,\ 3)$$

式中　　$e_K \to$（起点，终点）

显然　　$(1,\ 2) \neq (2,\ 1)$

通常图可化简为

$$G=(V,\ E)$$

如图1—3为

$$G=(V,\ E)$$
$$V=\{1,\ 2,\ 3,\ 4\}$$
$$E=\{(1,2),(2,3),(3,1),(1,4),(4,3)\}$$

而图1—7为

$$G=(V,\ E)$$
$$V=\{1,\ 2,\ 3\}$$
$$E=\{(1,\ 2),\ (3,\ 2),\ (1,\ 3)\}$$

在画图时两边相交，只要在相交处没有画黑点，那么点边的对应关系是不受影响的。例如，在图1—8中

图 1—8

图(a)中即使边a和b在图中是两线相交的，因为在相交处没有黑点，它们与点的对应关系还是

$$a \to (1,\ 3)$$
$$b \to (2,\ 4)$$

这就是说图a中只有两条边。可是在图(b)中，因为两线相交处画有黑点5，点边的对应关系就不同了，而有下列关系：

$$a \to (1,\ 5)$$
$$b \to (2,\ 5)$$
$$c \to (3,\ 5)$$
$$d \to (4,\ 5)$$

在我们对一个图下定义时，不一定要求组V和组E是有限的，但是要求组V不是空组。在理论和应用中，这些差不多都是有限的，有着有限点和有限边的图叫做有限图，否则就叫做无限图。图1—9是两个无限图的一部分。

图 1—9

以后除非明确说明是无限图外，我们所研究的都是有限图，所以以后所讲的线性图总是有限图，因而不再加有限两字了。

第四节　关联和次数

如果某一个点是一条边的两个端点中的一个，我们把这条边与这个端点的关系称为关联，或者说这条边是与这个点相关联的，或者说它是连接到这一个点的。例如，图1—10中边a，b和c是与点1相关联的，而a、d，e和f是与点2相关联的。

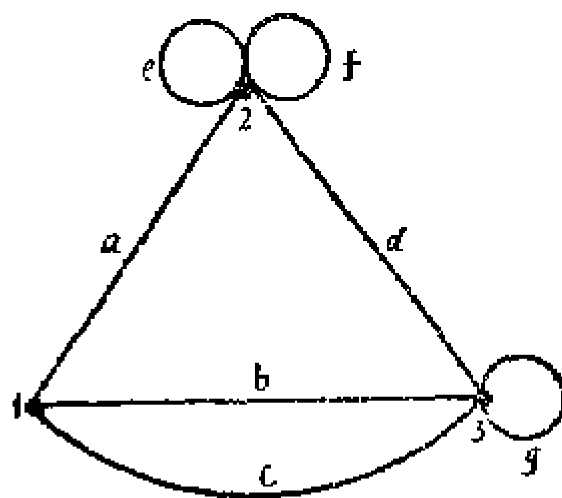


图　1—10

在图1—10中，边g的两个端点同是一个点3，这条边g就叫做自环，同样边e和边f也是自环。

在一对点之间可能连接一条以上的边。如图1—10中点1和点3之间有着两条边b和c，这两条边就称为平行边。

一个既没有自环，又没有平行边的图叫做简单图。例如图1—3和图1—6就是。

与每个点相关联或连接着的边数是图的一个很重要的数据。与一个点相关联的边数称为该点的次数。用数学方式来表达，可以$d(v)$代表v点的次数，$n(s)$代表与v相关联的自环数，n_n代表除自环外与点v相关联的边数，那么点v的次数为

$$d(v)=2n_s+n_n \tag{1—6}$$

例如在图1—10中，点1的次数为$d(1)=3$，这是因为这个点的$n_s=0$，$n_n=3$。点2呢，因为$n_s=2$，$n_n=2$，所以$d(2)=6$。同样的算法，$d(3)=5$。

假设边e关联着点p和q（就是说边e的两个端点为p和q）。当$p\neq q$（即分别为两个点）时，我们计算$d(p)$和$d(q)$都是1。当$p=q$（即p和q同为一点）时，边e（e是自环）的$d(p)$就算做2。这对图中每条边都是正确的，因此一个图中所有各点的次数总和为图G的边数的两倍，即

$$\sum_{i=1}^{n_v} d(v_i)=2n_e \tag{1—7}$$

式中$\sum_{i=1}^{n_v}$为图中所有点的总和，而n_e为图G中的边数，在图1—10中，共有7条边，即a、b、c、d、e、f、g，所以$n_e=7$。因此，$\sum_{i=1}^{n_v} d(v_i)=14$，这与上面计算的$d(1)+d(2)+d(3)=14$是相等的。

定理1—1：一个图中奇次数的点的数目总是偶数。

证：如果把一个图中奇次数和偶次数的点分开考虑，式（1—7）的左方可以分别写成为偶次数的点的次数与奇次数的点的次数和的总和，即

$$\sum_{i=1}^{n_0} d(v_i) = \sum_{e} d(v_j) + \sum_{o} d(v_K) \qquad (1-8)$$

式中$\sum_e$系偶次数的点的次数和，$\sum_o$系奇次数的点的次数和。根据式（1—7），式（1—8）的左方应等于偶数，而右方第一项也是偶数，那么第二项也必定是偶数（偶数之和是偶数），即

$$\sum_{o} d(v_K) = \text{偶数} \qquad (1-9)$$

因为式（1—9）中每一个$d(v_K)$是奇数，这个和项数一定是偶数才能使和成为偶数。▌

凡是一个图的各点的次数都相等（不管是偶数还是奇数），这种图就称正则图。

一个点而没有相关联的边的，称为孤立点。也可以说，一个点的次数为零的称为孤立点。例如，图1—11中的点6就是一个孤立点，它没有与之关联的边。

凡是次数为一的点称为悬挂点，也可以称为终点。关联到悬挂点的边称为悬挂边。图1—11中的点5就是悬挂点，而边e是悬挂边。

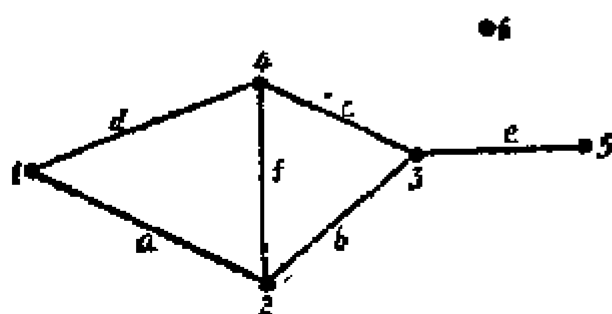


图 1—11

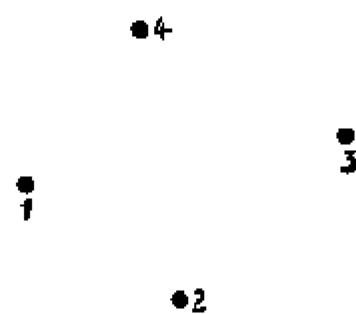


图 1—12

在第三节中，说到图的定义为$G=(V, E)$，其中边组E可能是空的或没有的。这种没有边的图就称为零图，有时也称为空图或空组或空集，符号为ϕ。换句话说，零图中的点都是孤立点。图1—12是一个有4个点的零图。零图虽然没有边但是不能没有点，所以图的点组V是不能等于ϕ的，否则就不成其为图了。所以，根据定义，一个图至少必须有1个点。有的图论书中，把既无边组又无点组的图称为零图，而把只有点而无边的图称为点图。

第五节　图论中的论证

逻辑中论证方式的核心是逻辑陈述。下面举三个等价陈述：

（一）如果H成立，那么C成立。

（二）$H \Rightarrow C$，符号$\Rightarrow$为蕴涵符，例如，$H \Rightarrow C$可以理解为H成立时，C必成立。

（三）H是C的充分条件。

其中H代表着作为假设的某种陈述，C是由H所得结论的陈述。例如

1. 如果甲是在珠穆朗玛峰，那么甲是在地球上的最高点。
2. 如果甲是在珠穆朗玛峰而不穿登山服，那么甲是很冷的。
3. 如果甲淹没在水中，那么甲便淹死。
4. 如果甲在经度$x°$，纬度$y°$，那么甲是在某城市中。

上述四种陈述在从假设而来的结论上是相似的，但是1.和4.不同于2.和3.。因为在1.和4.中，如果把假设与结果颠倒过来（即H与C相交换），就是把结论当作假设，那么把假设

作为结论也是成立的。这种可逆关系是以必要和充分条件为特征的，因此可用下面的不同的等价陈述：

（一）当且仅当H成立时，C成立。

（二）C成立的必要和充分条件是H。

（三）如果H成立，那么C成立，反之亦然。

（四）$H \Longleftrightarrow C$（$\Longleftrightarrow$称为等价符，两个命题H和C是等价的，就是说H和C都成立或都不成立）。

常常用于命题论证而不同于上述的两种陈述法的一种所谓否定陈述。如果把上面第一种陈述颠倒过来加以否定，则

（一）如果C不成立，那么H不成立。

（二）C不成立$\Rightarrow H$不成立。

（三）C不成立是H不成立的充分条件。

这种陈述是等价于上面第一种陈述的。如果这种陈述方式用上述的1.～4.作为例子，那么结果的否定陈述都是正确的。此外，必要和充分的陈述1.和4.对两种顺序（即假设与结论可以交换的）的否定陈述是正确的，而陈述2.和4.（充分的但不必要的）只对一种顺序的否定是正确的。在逻辑结构和论证技术中都假定着$H \Rightarrow C$，或C不成立$\Rightarrow H$不成立的任何一个蕴涵着另一个。因此，如果两者中任何一个可以证明是正确的，那么另一陈述也是正确的。

图论发展中的中心环节是以一个图或子图的某一或某些已知性质的逻辑陈述作为假设H，用它来证明一个图或子图的一个或更多的性质做为结论C。

论证技术的概要如下：

命题或逻辑陈述的基本形式为：

$H \Rightarrow C$，或$H \Longleftrightarrow C$。

（一）直接论证

1. 充分性的结论：假设了H的性质，从而证明了C的性质，就是假设H，证明C。
2. 必要性的论证：假设C，证明H。

（二）间接论证

1. 充分性的论证：假定C不成立，证明H不成立 。
2. 必要性的论证：假定H不成立，证明C不成立。

（三）反证法

这是为了证明某一命题是真而证明其反命题为谬的一种方法。

1. 充分性的论证：假定H成立和C不成立的条件，证明矛盾处。
2. 必要性的论证：假定C成立和H不成立的条件，证明矛盾处。

上面这三大类论证法都表明在论证中逻辑陈述所需完成的任务，但都没有表示出如何去达到这些逻辑陈述的论证。任何一个论证的最重要方面在于证明者的才能。下面四种方式可作为实现一个论证的指南。就是提供一些证明上述各种论证的方法。

（一）多重陈述法

根据已给假定的资料做出一系列的逻辑陈述来引导到所要证明的某一性质或某些性质。为了减少可能造成的误差，所做的陈述应该简短，当然必须是正确的。

（二）归纳法

1. 对于$n=1$或$n=N>1$，证明H（或C）$\Rightarrow C$（或H）。

2. 对于 $n>1$ 或 $n>N$，H（或C）$\Rightarrow C$（或H）。

3. 对于 $n+1$ 或 $N+1$，证明H（或C）$\Rightarrow C$（或H）。

（三）存在法

这种方法的论证也许是最令人信服的。从H（或C）的资料出发，用一个实际的结构图表来说明产生C（或H）。

（四）穷举法或称完全归纳法

这种方法只适用于只有很少几个（2个、3个或4个）逻辑陈述的情况下，而且这些情况可以分别处理的。由于所观察的情况缺乏把握，这个方法势必冗长，而且不够明确。

对论证技术的概述方面，最后可注意的是：某个命题的陈述的反例可用以证明该命题的不成立。

习　题

1—1　画出具有1个点、2个点、3个点、4个点的简单图。

1—2（1）画出一个代表2幢房子和3个公用事业问题的图。

（2）画出一个代表4幢房子和4个公用事业问题的图（四个公用事业为水、煤气、电及电话）。

1—3　命名10种情况（比赛、活动、实际生活等）把这些借助于图表表示，解释点和边代表什么。

1—4　画出惠斯登电桥电路的拓扑图。

1—5　有一个迷宫如图1—13所示，用一个图代表这个迷宫，使点表示通道或表示堵住的一端（如数字），边代表两点间一条可能的路径。

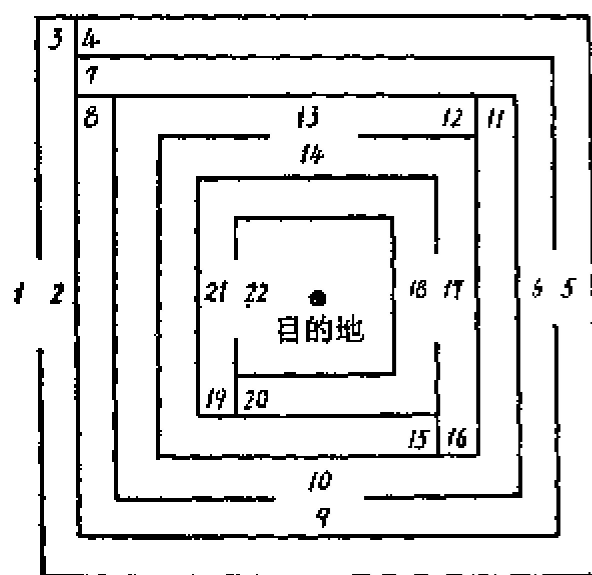


图 1—13

1—6　试确认具有n_v个点的简单图中任意一点的最大次数是n_v-1。

1—7　证明具有n_v个点的简单图中边的最大数目是$n_v(n_v-1)/2$。

1—8　5个朋友围着圆桌吃饭，希望每顿饭旁边坐着一个新邻居。问可能有几种排列？按顺时针次序标出每次排列的顺序。

1—9　画一个线性图（如果可能的话），在这个线性图中，点的次数是

（1）2，2，2，2，2。（2）2，3，4，5，6。（3）3，3，4，5，5，5，6。

1—10　试举出3个实例，说明可以用图论解决。

第二章　行走、路径和回路

第一节　行　　走

一个行走，有的书上称边列，也有的称链，是一系列边线，它具有下列特性：

（一）在系列中，除了第一条边和最后一条边外，其它边的一个端点是前一条边的端点，而另外一个端点是后一条边的端点。

（二）第一条边的一个端点是次一条边的端点，而另一个端点是起点。

（三）最后一条边的一个端点是前一条边的端点，而另一个端点是终点。

（四）每一条边只用一次。

当一个行走的起点和终点是不同的，起点和终点的次数都等于一，这个行走称为开行走，否则称为闭行走。以图 2 — 1 为例，图(a)是一个线性图。

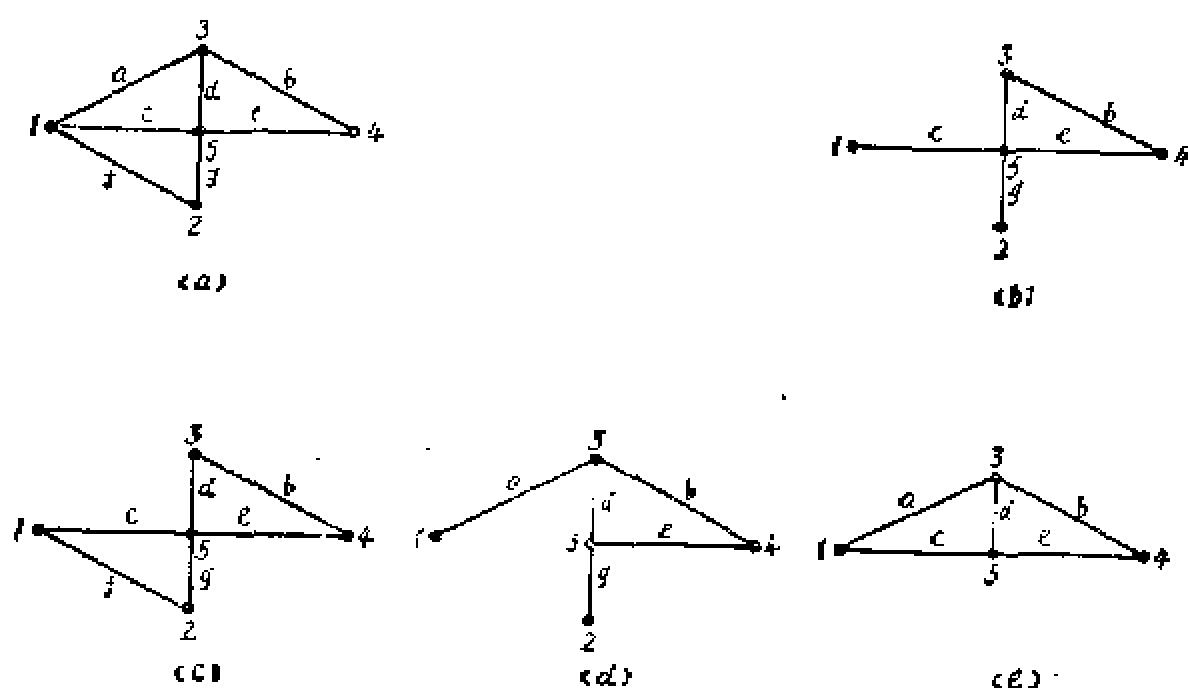


图　2—1

图（b）中的行走（c，d，b，e，g）是一个开行走，它的起点为1，而终点为2，这两个点不是同一点。图（c）中的行走（c，f，g，d，b，e）是一个闭行走，因为点5既是起点，又是终点，同为一个点。图（d）中的边组(c，d，e，b，d，g)不是一个开行走，也不能称为一个闭行走，而只能叫做一个边组。这是因为虽然起点1和终点2不同，但是边d却用了两次，这是不符合行走定义的第四条特性的。同样，图（e）中的边组（a，c，d，b，e，d）既不是一个闭行走，也不是一个开行走。理由是虽然起点3和终点3同为一点，但是边d却用了两次；若以3为起点，以5为终点，而起点和终点的次数都不等于1。由此可见，一个行走中的每条边只能用一次，而点呢？是可以通过一次以上的。

行走的符号用W，例如，行走（c，d，b，e，g）可以写作W（c，d，b，e，g）。

在图2—1中，图（b）和（c）各称为对应一个行走的线性图。

第二节 路 径

所谓路径是点p和点q之间的一个开行走中所有边的边组，它们满足下列两个条件：

（一）起点和终点分别为p和q。

（二）点p和q的次数各为1，而其它各点，即中间点的次数为2。当然，这里的次数只计算路径中所涉及的各边，而不是路径所在的整图中的所有边。例如，在图2—1中，对应行走（a，d，e）的线性图，如图2—2（a）所示，因为符合路径定义的两个条件，所以这是点1和点4之间的一条路径；但是行走（a，d，g，f，c，e），它的对应图如图2—2（b）所示，不能作为点1和点4之间的一条路径。这是因为它虽然符合路径的第一条件，即起点和终点分别为1和4，但是它不符合第二条件，即这里起点1的次数为3，并且中间点5的次数为4。注意一个行走中可以包括自环，而一条路径却不能有自环的。例如，图2—3（a）中粗线所示的边组（a，b，c，d）是一个开行走，而图2—3（b）中粗线所示边组（a，b，d）是一条路径。

图 2—2

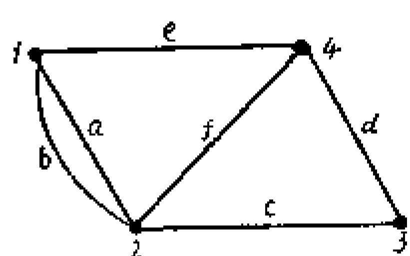
图 2—3

点i和点j之间的路径符号用P_{ij}，例如图2—2（a）所示的路径可以写作$P_{14}=(a,d,e)$。

在一个图G中，如果在每一对点之间至少有一条路径的，或者换句话说，我们可以从图G中任何一个点沿着一条边或几条边到达任何其它点的，称为连通图；否则，如果有两个点而其间没有路径的称为分离图。例如在图2—2中，各图的任何两点之间有一条或几条路径；所以都是连通图。在图2—4中有两个连通图，两者之间没有连通的边，所以是没有路径的，这个图就是分离图。在图1—11中有孤立点6，它与其它点之间没有路径，所以也是分离图。不止一个点的零图，如1—12也是一个分离图。

如果一个图G中有这样一个点p，它可以截割开来使原图分离但不产生孤立点的，G就称为可分图。这个点p就叫做截割点。例如，图2—5（a）中的点3就是一个截割点，把点3截割后，图（a）就分成两个图，如图（b）所示。

凡是不能截割一个点而分离开来的图称为不可分图。图 2 — 6 就是一个例子



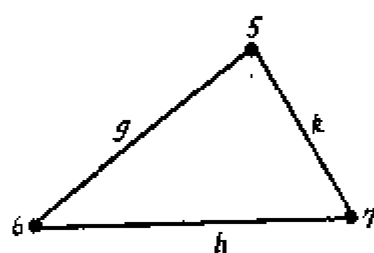
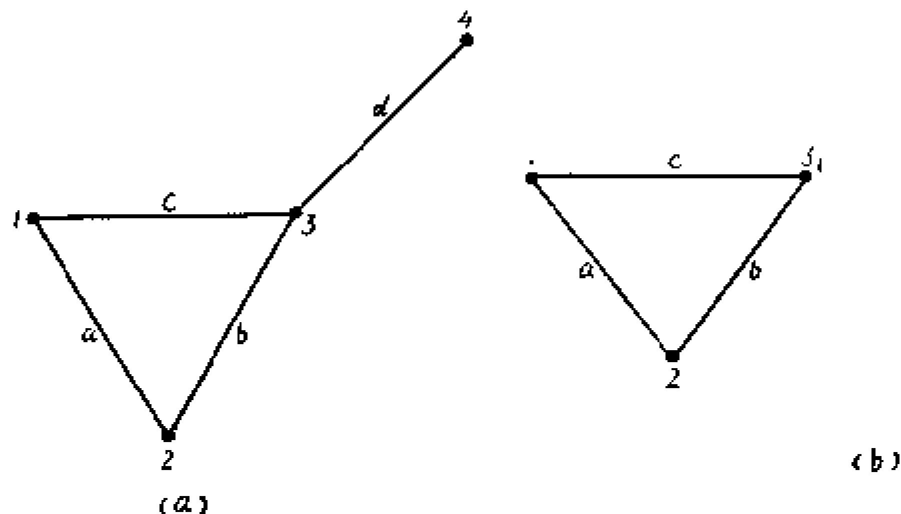


图 2 — 4



(a)　　(b)

图 2 — 5

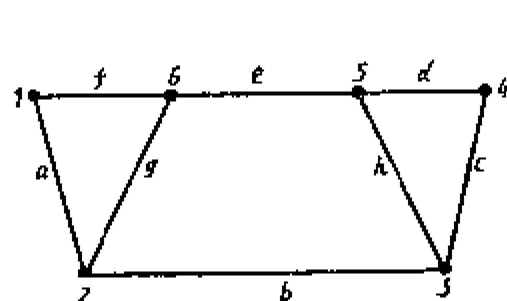


图 2 — 6

第三节 子 图

如果一个图g中所有点和边都属于一个图G的，而g的每一条边的端点正是G中的同一端点，那么 g 称为G的子图。例如，图 2 — 7 (a)正是图 2 — 6 的一个子图。同样，图 2 — 1 中的 (b)，(c)，(d)和 (e)，以及图 2 — 2 中的 (a) 和 (b) 都是 2 — 1 (a) 的子图。但是图 2 — 7 (b) 却不是图 2 — 6 的子图，这是因为图(b)中的点 3 和边 b 和 c 都不在图 2 — 6 的对应位置中。

子图的概念是和集合论中的子组或子集的概念相类似的。所以这里就用集合论中的符号 $g \subset G$ 来表达"g" 含子G" 或g是G的子图。反过来，G中包含着g可以用符号$G \supset g$。

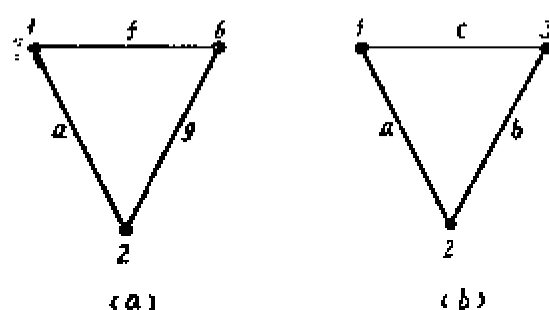
根据子图定义得到下列性质：

（一）每个图是它自己的子图。

（二）G的子图的子图也是G的子图。

（三）G中的一个单一点是G的子图。

（四）G中的一条单一边连同它的两个端点也是G的子图。



(a)　　(b)

图 2 — 7

既然一个图就是它自己的子图，因此与原图不同的子图常称为真子图。真子图包含于图G之中，记为$g \subset G$，而子图g包含于G中记为 $g \subseteq G$，这是因为子图中可以包含G本身。在以后我们讲到的子图差不多都是真子图，我们就简单地称为子图了。

组成分离图的子图称为部分，部分本身也是一个线性图，而且是一个连通图，所以有的书中叫做最大连通子图。孤立点也可以看做是一个子图。

图G的两个（或更多）子图g_1和g_2如果没有任何共同的边，那么这两个子图就称为边不共用子图。例如图 2 — 7 (a)和图 2 — 8 是图 2 — 6 的两个边不共用的子图。这里可以看

出，虽然边不共用的子图没有任何共同边，但是可以有共同点。象这里所举的例子中两个边不共用的图却有着共同点 2 和 6。如果两个子图非但没有任何共同边，又没有任何共同点，那么这种子图就称为点不共用子图。很明显如果两个子图没有共同点，当然就没有共同边了。例如图 2 — 7（a）和图 2 — 9 是图 2 — 6 的两个点不共用的子图。

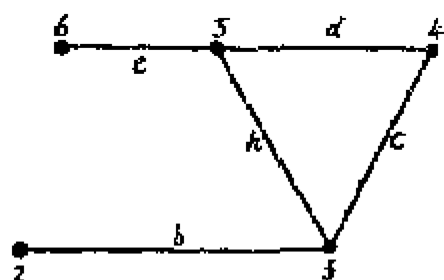


图 2 — 8

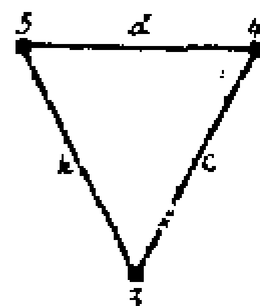


图 2 — 9

第四节 回　路

一个闭行走中各点的次数都为 2 的称为一个回路。例如，图 2 — 7 中的两个图都是闭行走，又是回路。再看图 2 — 1（c），它的确是一个闭行走，但是因为点 1，2，3，4 的次数虽然都是 2，可是点 5 的次数却是 4，所以不是一个回路。

自环也是一种回路。虽然自环只有一个点，但是这点的次数是 2。

回路的符号为 C。例如，在图 2 — 7 中，图（a）的回路可以用 C（a，g，f）来表示，图（b）的回路可以用 C（a，b，c）来表示。

凡是一组回路而其中任何两者之间没有共同边的，称为回路的边不共用并集。

现在我们可以这样讲，一个闭行走中的所有边或者是一个回路，或者是一个边不共用回路的并集。正如图 2 — 1（c）的行走（c，f，g，b，d，e）不是一个回路而是 C（c，f，g）和 C（b，d，e）的边不共用回路的并集。可能有这样的一个边组，把这个边组分解成回路的方法不止一个。可是，只要有一个分解方法把这边组分解成若干回路，而其中任何两者之间没有共同边的，那么这个边组就是边不共用回路的并集。例如，图 2 —10（a）的边组（a，b，c，d）是边不共用回路的并集，但是图（b）的边组（a，b，c，d，e）却不是边不共用回路的并集，因为这个边组不能分解成任何两者之间都没有共同边的回路。

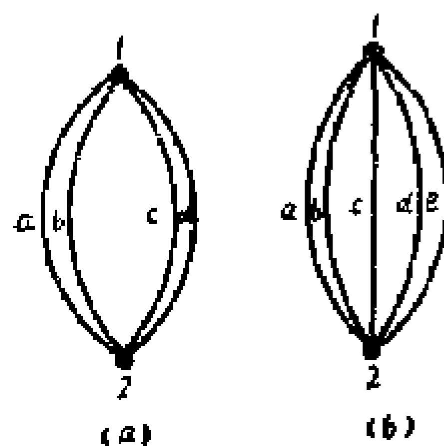


图 2 —10

第五节 同　构

如果两个图 G_1 和 G_2 之间在它们的点和边的关联方面保持着一对一的对应关系，这两个图就叫做同构图，或称它们是同构。例如，在图 2 —11中，图（a）和图（b）是同构的。它们的对应关系是：图（a）中的点 1，2，3，4 和 5 分别对应着图（b）中的点 1′，2′，3′，4′和5′；图（a）中的边 a，b，c，d，e 和 f 分别对应着图（b）中的边 a'，

b'，c'，d'，e'和f'。所以同构的图除了编号不同外，其实是同一图的不同画法。例如图2—12中的两个图是同一图的不同画法，也是同构的。

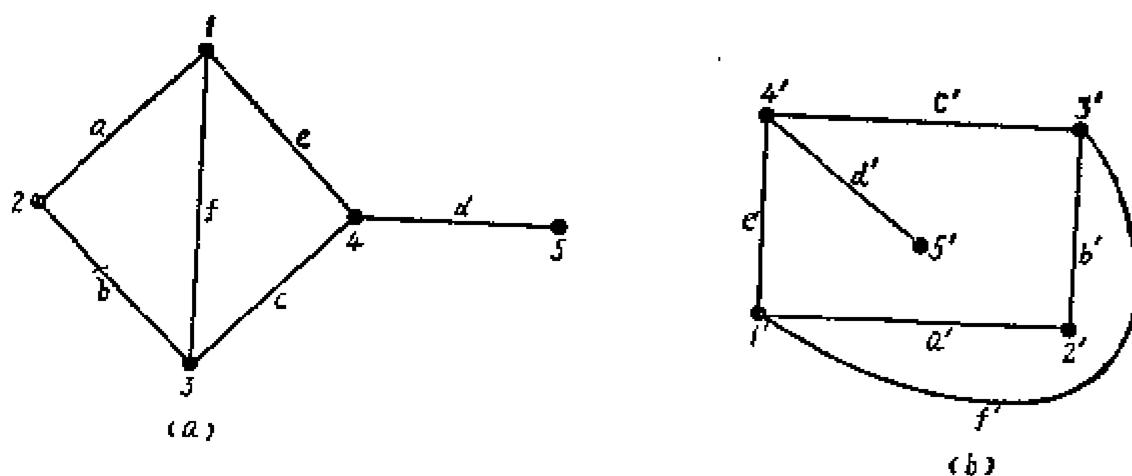


图 2—11

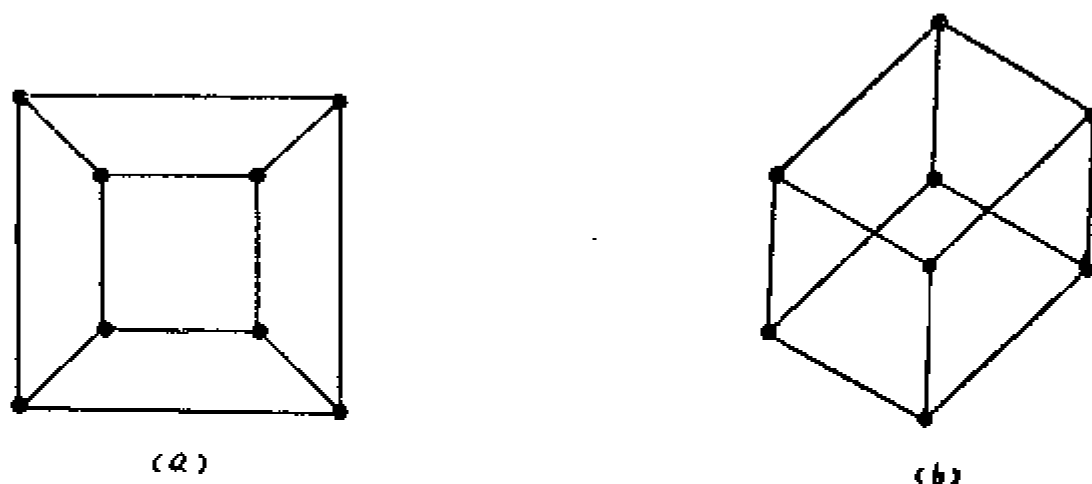


图 2—12

判断两个图是否同构并不常是一件容易的工作。例如，图2—13中的3个图是同构的，但是只是粗略地看看是看不出来的。我们一定要把点和边都编好号，再观察点和边之间对应关系和关联关系，才能明确地判明它们的同构性。

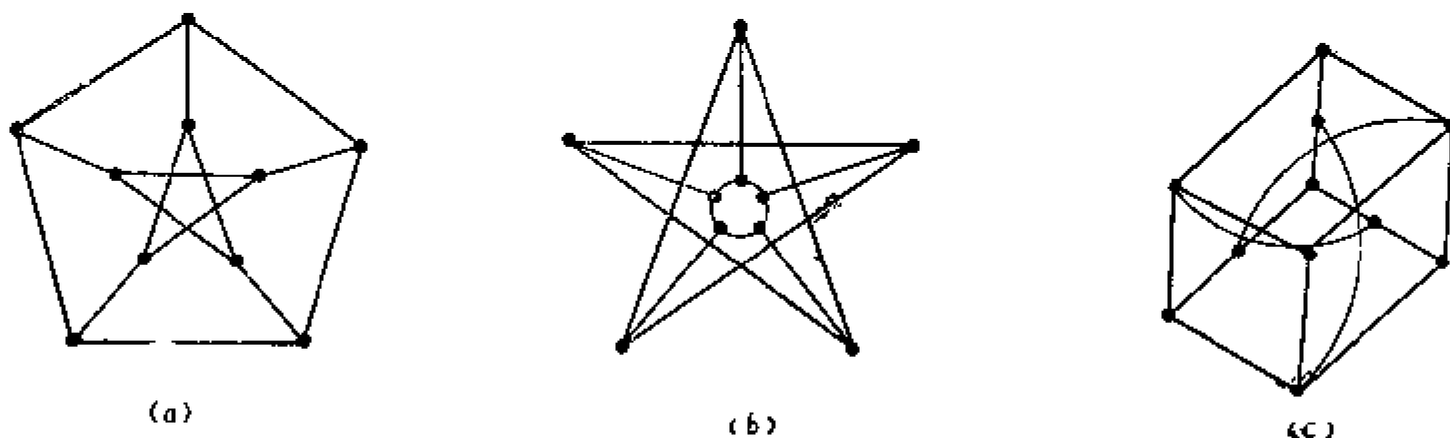


图 2—13

根据同构性的定义，两个同构图必须有：

（一）相同的点数；

（二）相同的边数；

（三）有相等的同次数的点。

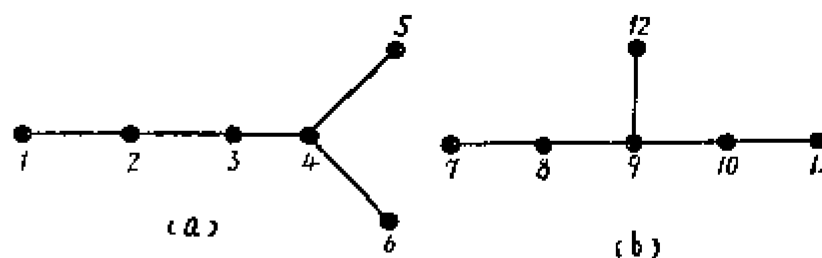


图 2—14

可是这三条只是同构的必要条件，而不是它的充分条件。例如图2—14中的两个图都是能满足上述三个条件，可是两图不是同构的。这是因为如果两图是同构的话，那么图（a）

中的点4必须对应于图（b）中的点9，因为两图中没有其它点的次数为3。但是在图(a)中，关联到点4的有两个悬挂点5和6，而在图（b）中，关联到点9的只有一个悬挂点12。

在图论中，检查同构性的既简单又有效的判断准则仍是一个有待解决的重要问题。

第六节 图的运算

象大多数的数学一样，大的图形可以考虑是小图的组合，并且从小图中来推导它们的性质。既然从第一章的开头就讲，线性图是用点组和边组（或称为点集和边集）来下定义的，这就很自然地可以采用集合论中的组运算或集运算来做线性图的运算。

两个图G_1和G_2分别为$G_1=(V_1,E_1)$和$G_2=(V_2,E_2)$，它们的并集为另一个图G_3。如用$\cup$作为并集的符号，那么上述两个图的并集可以写作$G_3=G_1\cup G_2$，它的点组$V_3=V_1\cup V_2$，和它的边组为$E_3=E_1\cup E_2$。换句话说，G_3是包括G_1或G_2中所有的点和边的图。

G_1和G_2的交集是另一个图G_4。如用符号$\cup$代表交集，那么$G_4=G_1\cap G_2$。G_4只包括G_1和G_2中两者共有的边，即G_1和G_2的公有点和边。

G_1和G_2的环和，符号为$\oplus$，是一个图G_5，写作$G_5=G_1\oplus G_2$。G_5的点是$V_1\cup V_2$，而它的边是G_1或G_2中所特有的，而不是两者都有的边。

G_1和G_2的差集，符号为$\ominus$，是另一个图G_6，写作$G_6=G_1\ominus G_2$。G_6是从G_1中删去G_2中也有的各边所余留下来的图。同样$G_7=G_2\ominus G_1$，G_7是从G_2中删去G_1中也有的各边所余留下来的图。

从上述各种运算的定义中可知

$$G_1\cup G_2=G_2\cup G_1 \qquad (2-1)$$

$$G_1\cap G_2=G_2\cap G_1 \qquad (2-2)$$

$$G_1\oplus G_2=G_2\oplus G_1 \qquad (2-3)$$

另外

$$G_1\ominus G_2=(G_1\cup G_2)\ominus(G_1\cap G_2)=(G_1\ominus G_2)\cup(G_2\ominus G_1) \qquad (2-4)$$

这里我们虽然应用了集合论中的集运算，但是我们所研究的是图，所以用图来表示这种运算时，就如例2—1所示。

例2—1　在图2—15中(a)中给出两个图G_1和G_2，G_1和G_2并集的$G_3=G_1\cup G_2$见图（b），G_1和G_2交集所得$G_4=G_1\cap G_2$见图（c）。G_1和G_2环和所得的$G_5=G_1\oplus G_2$见图（d），图（e）中的差集$G_6=G_1\ominus G_2$，而差集$G_7=G_2\ominus G_1$则见图（f）。

如果G_1和G_2是边不共用的，那么$G_1\cap G_2$是个零图，并且$G_1\oplus G_2=G_1\cup G_2$，如果$G_1$和$G_2$是点不共用的，那么$G_1\cap G_2$就完全是空的，即既无边，又无点。

任何一个图G

$$G\cup G=G\cap G=G \qquad (2-5)$$

$$G\oplus G=\phi \qquad (2-6)$$

其中，ϕ是零图或空组。

如果g是G的子图，那么根据定义$G\oplus g$是从G中删去g中所有的边之后所余留下来的那个子图。所以$G\oplus g$可以写成$G\ominus g$，而g总是在G之中，即$g\subset G$。由于这种补余的性质，所以$G\oplus g=G\ominus g$常常称为G中g的余图用符号$\overline{g}$表示。例如在图2—16中，（a）是一个图

G，（b）是G的一个子图g，（c）是子图g的余图$\overline{g}$，即$G\ominus g=\overline{g}$。

利用图的运算，我们可以得到很多重要的结果。

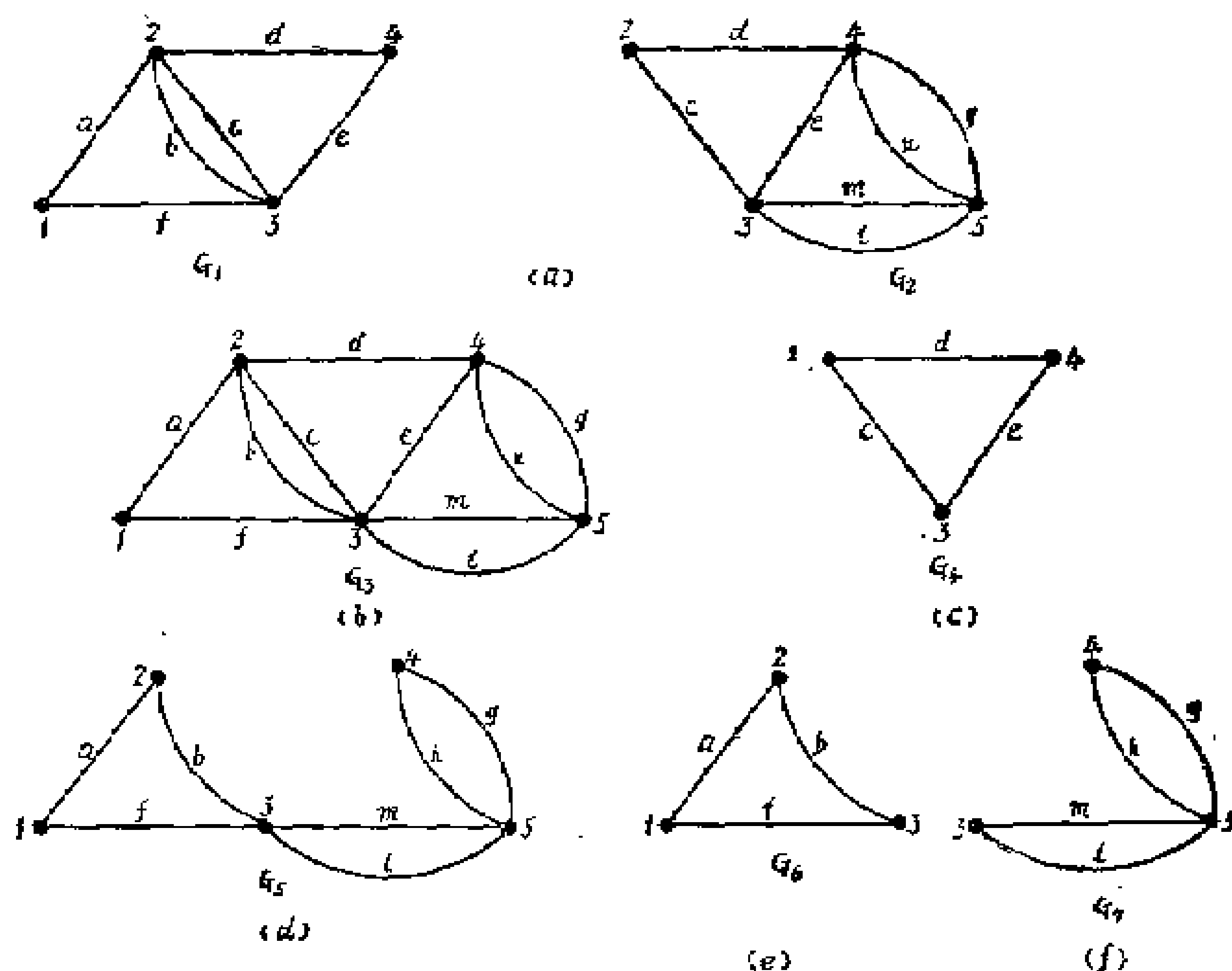


图 2—15

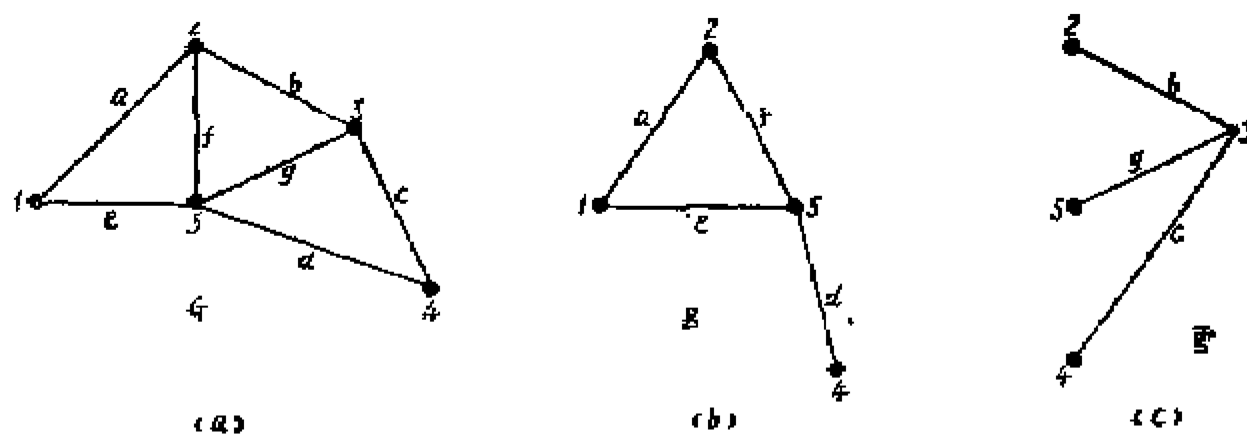


图 2—16

第七节 图的分解、删除、合并和短捷

如果一个图G和它的子图g_1和g_2有下列关系：

$$g_1\cup g_2=G \tag{2—7}$$

和

$$g_1\cap g_2=\phi \tag{2—8}$$

那么说图G分解为两个子图g_1和g_2。换句话说，图G中每条边不是在子图g_1中就是在子图g_2中，但不会在g_1和g_2中都出现。可是有些点可能在g_1和g_2中都出现。例如，图2—16（a）分解成图（b）和（c）两个子图。图的分解不限于分解成两个子图，有时也可以分解成若干个子图，只要这些子图是边不共用的，并且共同包含原图中每一条边。在分解时，孤立点是不管的。

如果在一个行走中所有边的边组，它们的起点和终点分别为p和q，而且不是p和q之

间的一条路径，那么这个开行走，可以分解成为一个开行走和一个闭行走，其中（1）原来开行走中的每条边正是现在分解成的两个行走中的边，和（2）分解出的开行走中的所有边的这一个边组是点 p 和 q 之间的一个路径。

例如，图 2－2（b）的开行走，上面已经说过不是点 1 和点 4 之间的路径。现在我们把这个行走分解成一个开行走如图 2－2（a）所示，和一个闭行走如图 2－17 所示。原行走（a，d，g，f，c，e）中的各边正是图 2－2（a）和 2－17中的各边，而开行走（a，d，e）正是点 1 和点 4 之间的一条路径。象上面所说的开行走也可以分解成若干条路径。如图 2－2（b）可分解成$P_{14}=(a,d,e)$，$P_6=(c)$和$P_{15}=(f,\ g)$。同样，一个闭行走中的所有边如果不是一个回路的话，我们也可以用分解法把它分解成若干个闭行走，其中（1）原来闭行走中的每条边正是现在分解出来的若干个闭行走中的边，和（2）每一个分解出来的闭行走中所有边的边组是一个回路。

例如，上面提到的图 2－1（c）就是这里的图 2－18（a），是个闭行走，但不是一个回路。如果我们把它分解成两个闭行走，如图 2－18（b）和（c）所示，这两个子图就都是回路。

一个闭行走也可以分解成为若干条路径，如图 2－1（c）可分解成$P_{14}=(c,e)$，$P_{12}=(f,g)$和$P_{54}=(d,b)$。

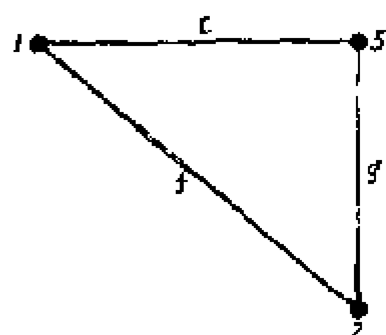
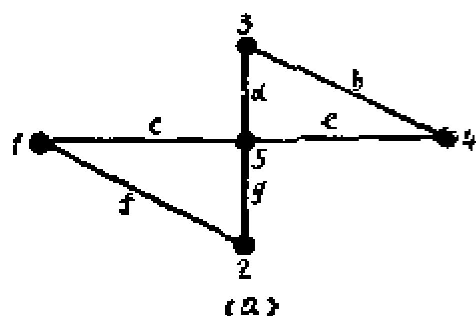
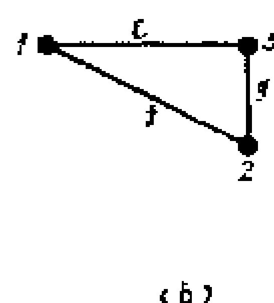


图 2－17





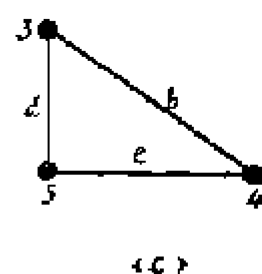
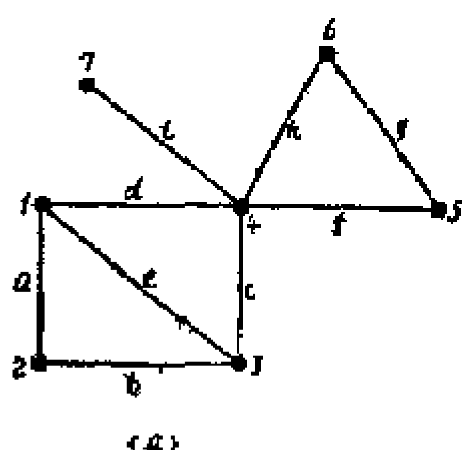
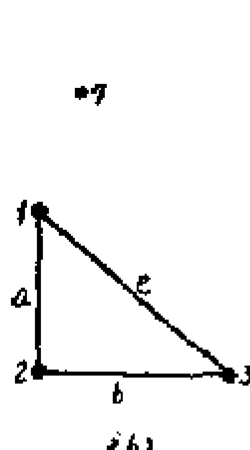


图 2－18

删除就是从图中移去一些点或边。这与上一节所讲的差集是同一种运算的两种不同称法。如果点v_i是图G中的一个点，删除v_i，即$G\ominus v_i$，就是从G中移去v_i后所得到的一个子图。其实，删除一个点常常意味着也删除掉与这个点相关联的各边。但是删除一条边e_j并不意味着也删除它的端点。所以$G\ominus e_j=G\oplus e_j$。例如，图 2－19（a）是一个线性图，当我们删除点 4 时，与点 4 相关联的边 c，d，f，h，l 也都删除了，所余留的子图就如图（b）所示。但是当删除边 b 时，它的端点 2 和 3 仍然存在，所以余留的子图就如图（c）所示。





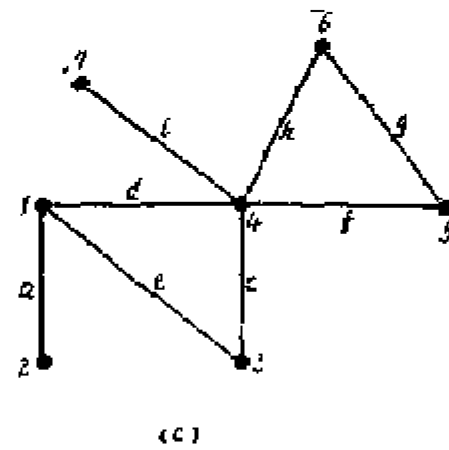


图 2－19

合并也可以称为重叠，是指把原来关联到这两个点或其中任何一个点的边都关联到这个

新点。所以两个点的合并不变更图的边数而只将图的点数减少1。例如，在图2—19（a）中，如果把点2和点3合并，则成图2—20。这里两个点合并成为一个新点（2，3），而边b则成为一个自环。图2—19（a）共有7个点和9条边。把点2和点3合并后，图2—20只有6个点，但仍有9条边，即只减少了1个点，而不影响边数。

短捷是指把一条边移去，同时把它的两个端点合并在一起成为一个点。所以短捷不同于删除一条边，因为前者删除的只是边而不影响它的端点，但是也不同于合并两个点，因为合并点时，边仍保留，而短捷一条边时，边就没有了。由此可见，短捷一条边时，同时把图的点数和边数都减少1。例如在图2—19（a）中，如果我们把边b短捷，则成图2—21，这里边b不再存在，而点2和点3则合并成一个新点（2，3）。所以图2—21只有6个点和8条边，点数和边数都比图2—19（a）减少1。

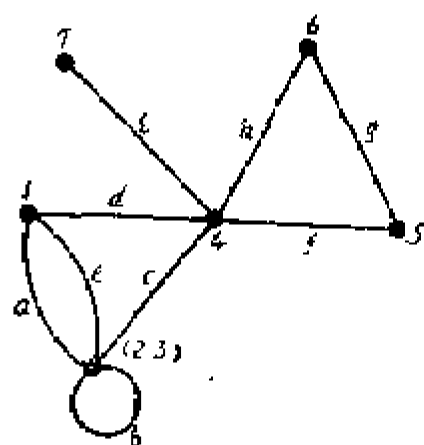


图 2—20

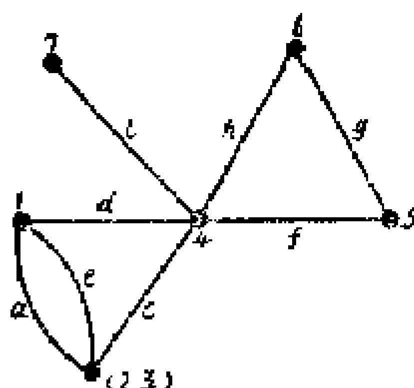


图 2—21

第八节　欧　拉　图

欧拉在解决哥尼斯堡桥的论文中，提出并解决了一个更加一般性的问题，在什么形式的图G中可以找出通过G中每条边确实只有一次的闭行走呢？这种闭行走现在称为欧拉线，有时也叫欧拉迹。而包括这个行走的图就称为欧拉图。更正式地讲：如果一个图G中某一闭行走包括G中所有边，这个行走就是欧拉线，这个图就是欧拉图，欧拉图的符号用E。

根据定义，行走总是相连通的。既然欧拉线也是一个行走，它包含图中所有边，一个欧拉图总是相连通的。有些图可能有些孤立点，可是孤立点对我们理解欧拉图并无什么意义，所以此后假定欧拉图中是没有孤立点的。因此欧拉图总是连通的。

现在我们要介绍一个重要的定理，它能立即告诉我们一个图是否是一个欧拉图。

定理2—1，当且仅当一个连通图G的所有点都是偶次数时，这个图就是欧拉图E。

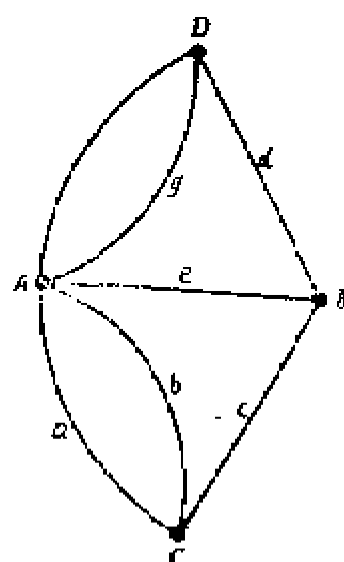
证：先假设这个图G是一个欧拉图E，那么它包括一条闭行走的欧拉线。当我们沿着这个行走观察时，每次遇到一个点v时，就有两条边关联到这个点v，一条是进来边，另一条是出去的边。这不仅适用于行走中的各个中间点，而且也适用于起点和终点。实际上就是一个点，我们开始自这个点出去，最后又进入这个点。所以如果G是一个欧拉图E，每点的次数是偶数的。

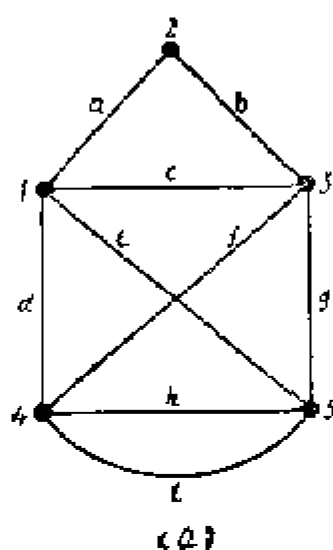
为了证明这个条件的充分性，假设G中各点都是偶次数，现在我们从任意点v组成一个行走W，并且通过G中各边，其中没有一条边是用过一次以上的。我们继续沿着这个行走前进，既然每点是偶次数的，我们可以从任意进入的点再出来，一直到达点v。既然v也是偶次数的，我们沿着行走走到终点时将最后到达v。如果这个闭行走W中包括G中所有的边，

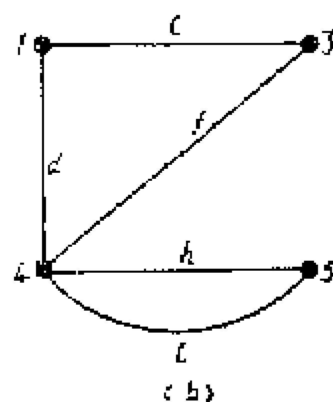
那么这个G就是欧拉图。如果不是的话，我们从G中删除W中所有的边而得到一个由G中余留下来的边所组成的子图g，既然G和W两者的所有点是偶数的，g的各点的次数也是偶次数的。而边，因为G是连通的，g和W至少在一个点a上是连通的。从点a开始，我们可以在图g中再组成一个新行走。既然g中所有的点是偶次数的，在g中的这个行走一定以点a为终端。不过在g中的这个行走可以与W相联合成一个新行走，它是以点v为起点和终点，并且比W的边数为更多些。这个过程可以重复几次，一直到通过G中所有的边而得到一个闭行走为止。因此G是一个欧拉图E。

现在让我们回过头来看看第一章所讲到的哥尼斯堡的7桥问题，欧拉把点代表陆地和岛，而把边代表桥，把图1—1画成一个线性图如图2—22所示。由图可见，在这个图中所有的点都不是偶次数的，因此，它不是一个欧拉图，所以一个人不可能从7座桥中的某一座桥开始，一次走遍7座桥而回到开始的桥，而每座桥又只能通过一次。

根据欧拉图的定义，如果一个图G中有一闭行走包括G中所有边，这个图就是欧拉图，现在让我们观察一个欧拉图。







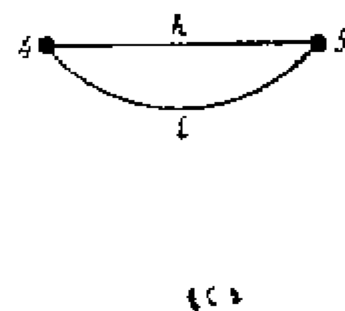


图 2—22　　　　图 2—23

例2—2　有一个欧拉图E如图2—23(a)所示。从1开始，我们得到一个闭行走（a，b，g，e），这也是一个欧拉图E_1，应用差集运算，$E\ominus E_1$如图（b）所示，其中$E_1=(a, b, g, e)$。假设在图（b）中我们用边c，d和f组成一个闭行走（c，d，f），这也是一个欧拉图E_2。所以$E\ominus E_1\ominus E_2$就是图（c）所示，其中$E_2=(c, d, f)$。最后的图（c）又是一个闭行走（h，l），而且也是一个欧拉图$E_3=(h, l)$，因此，图（a）的欧拉图有3个子图，也是欧拉图。

一般地讲，一个由闭行走组成的线性图分解成欧拉图的子图时有不同的分解法。例如在例2—2中，我们把图2—23(a)分解为$E_1=(a, b, e, g)$，$E_2=(c, d, f)$和$E_3=(h, l)$作为子图。但是，我们也可以把图（a）分解为$E_4=(a, b, c)$，$E_5=(c, e, g, h, l)$和$E_6=(c, e, g)$作为子图。欧拉图的一个重要特性是一个线性图中的所有欧拉图可以由几个单元欧拉图的组合来形成，这几个欧拉图称为生成元。现在我们先来研究两个欧拉图的环和。

设欧拉图E_1和E_2是一个线性图的两个子图，取它们的环和$E_1\oplus E_2$，其中$E_1\neq E_2$。为了找出$E_1\oplus E_2$中每个点的次数，设e_{11}，$e_{12}\cdots e_{1r}$是E_1中的一些边，它们关联到点p。又设e_{11}，$e_{22}\cdots e_{2s}$是E_2中的一些边，它们也关联到点p。无损于普遍性，我们可以设$e_{11}=e_{21}$，$e_{12}=e_{22}\cdots e_{1t}=e_{2t}$，但是$e_{1t+1}$，$e_{1t+2}$，…，$e_{1r}$和$e_{2t+1}$，$e_{2t+2}$，…，$e_{2s}$都是不同的。所以在$E_1\oplus E_2$中只有那些边$e_{1t+1}$，$e_{1t+2}$，…，$e_{1r}$，$e_{2t+1}$，$e_{2t+2}$，…，$e_{2s}$是关联到点$p$的。

因此，在$E_1 \oplus E_2$中并联到点p的次数是$r+s-2t$。既然E_1和E_2是欧拉图，r和s是偶数，所以$r+s-2t$是偶数。因此$E_1 \oplus E_2$也是一个欧拉图。例如在图2—15(a)中，因为G_1和G_2是欧拉图，所以在图（d）中的$G_1 \oplus G_2$也是一个欧拉图。因此我们得到下面定理。

定理2—2：设欧拉图E_1和E_2是一个图中的两个子图，那么$E_1 \oplus E_2$是一个欧拉图。

这个定理的普遍化是下一定理

定理2—3：设$E_p(p=1,2,\cdots n)$是一个图中的一些欧拉图，那么$E_1 \oplus E_2 \oplus \cdots \oplus E_n$是一个欧拉图。

为了方便起见，我们用符号$\{E\}$来表示一个线性图中所有可能的欧拉子图的集体。

有了定理2—2和2—3我们可以看到集体$\{E\}$满足下列条件：

（一）如果E_i和E_j是在$\{E\}$之中，那么$E_i \oplus E_j = E_j \oplus E_i$是在$\{E\}$之中。

（二）当有$E_0=\phi$时，那么在$\{E\}$中任何一个E_i，$E_0 \oplus E_i = E_i$ （2—9）

（三）对于$\{E\}$中任何一个E_i，

$$E_i \oplus E_i = E_0 = \phi \qquad (2—10)$$

（四）对于$\{E\}$中任何E_i，E_j和E_K，

$$E_i \oplus (E_j \oplus E_K) = (E_i \oplus E_j) \oplus E_K \qquad (2—11)$$

为了了解生成元的作用，我们以图2—24为例。这个图总共包含7个欧拉子图E_1，E_2，E_3，E_4，E_5，E_6和E_7。如图2—25所示。这些子图就是生成元，由生成元通过环和运算可以得到其它欧拉图。在图2—25中，由生成元E_1，E_2和E_3可以用环和运算得到E_4，E_5，E_6和E_7，即

$E_1 \oplus E_2 = E_7$　　$E_1 \oplus E_3 = E_4$

$E_2 \oplus E_3 = E_4$　　$E_1 \oplus E_2 \oplus E_3 = E_6$

下面是关于欧拉图的另一个重要定理。

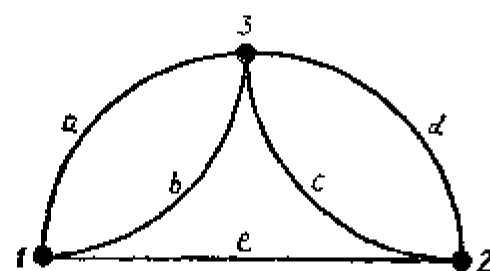
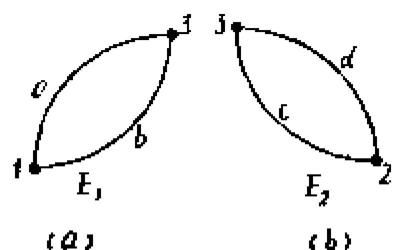
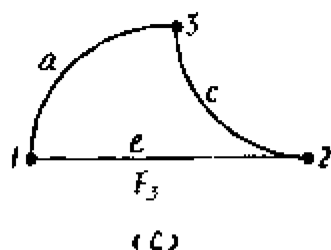
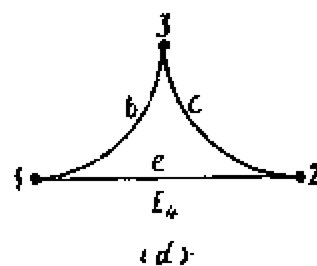
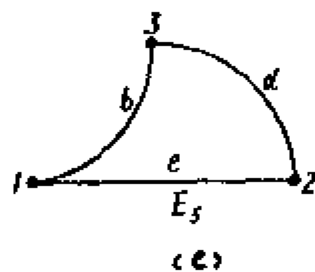


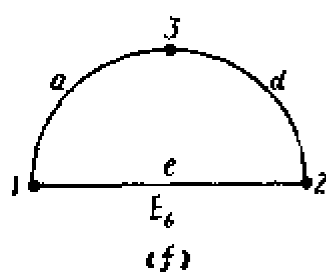
图 2—24











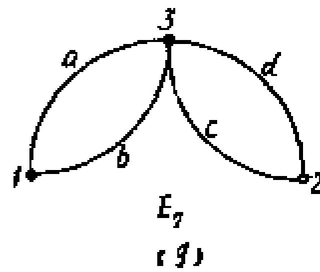


图 2—25

定理2—4：当且仅当一个连通图可以分解成若干回路时，这个图是一个欧拉图。

证：假使图G可以分解成若干回路，就是说，G是一个边不共用回路的并集。既然在一个回路中每点的次数为2，所以G中各点的次数为偶数。因此这个图G是一个欧拉图。

相反地，设G是一个欧拉图，取其中一个点v_1，至少有两条边关联到这个点v_1。设其中的一条边是在v_1和v_2之间，因为点v_2也是偶次数，一定至少有另外一条边，比如说连接在v_2

与v_3之间。这样推论下来，最后我们达到一个以前已经到过的点，因此形成一个回路C。设从G中删除这个回路C。所余留下来的图（不一定是连通的）中的所有点一定仍然是偶次数的。从这个余留下来的图中，仍照上面所说的从G中删除一个回路C那样再删除另一个回路。这样继续下去，最后一条边也不剩。这就是说这个图G可以分解成若干个回路。

定理 2 — 5： 如果P_1和P_2是同一对点之间的两条路径，那么$P_1 \ominus P_2$为欧拉图。

证：设P_1和P_2是一个图G中点i和j之间的两条路径，我们从点i和j为端点插入一条边y组成一个新图G'，注意P_1和P_2也是G'中的两条路径。$P_1 \cup y$是P_1中的所有边和边y的一个边组。$P_2 \cup y$是P_2中的所有边和边y的一个边组。很明显这些边组是两个欧拉图。达两个边组的环和

$$\{P_1 \cup (y)\} \ominus \{P_2 \cup (y)\} = P_1 \oplus P_2 \qquad (2-12)$$

很据定理 2 — 2，两个欧拉图的环和的结果也是一个欧拉图。因为$P_1 \oplus P_2$中的所有的边是在G之中，所以$P_1 \oplus P_2$是G中的一个欧拉图。

例如，在图 2 —24中的线性图G中，设$P_1=(a,d)$，$P_2=(a,c)$。那么$P_1 \cup (y)$和$P_2 \cup (y)$如图 2 —26(a)和(b)所示。注意这两个图都是欧拉图。同时注意边y不在G之中，$P_1 \oplus P_2=(c,d)$是G中的一个欧拉图。

图 2 —26

把这个定理普遍化得下一定理。

定理 2 — 6： 如果P_r是同一对点之间的若干条路径，其中$r=1, 2, \cdots\cdots, 2k$，那么$P_1 \ominus P_2 \ominus \cdots\cdots \ominus P_{2k}$（偶数条路径的环和）是一个欧拉图。

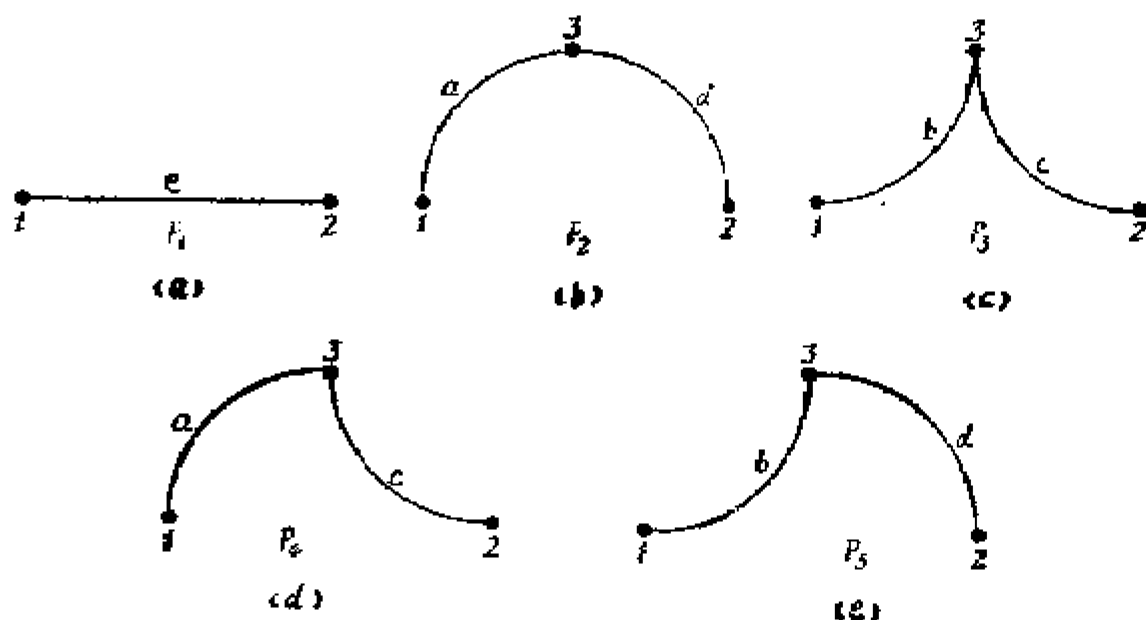


图 2 —27

证：我们有

$$P_1 \oplus P_2 \oplus \cdots\cdots \ominus P_{2k} = (P_1 \oplus P_2) \oplus (P_3 \oplus P_4) \oplus \cdots \oplus P_{2k-1} \oplus P_{2k} \qquad (2-13)$$

因为 $P_{i-1}\oplus P_i$ 根据定理 2 — 5 是一个欧拉图，而且据定理 2 — 3，几个欧拉图的环和是一个欧拉图。所以式（2 —13）的右方是一个欧拉图。

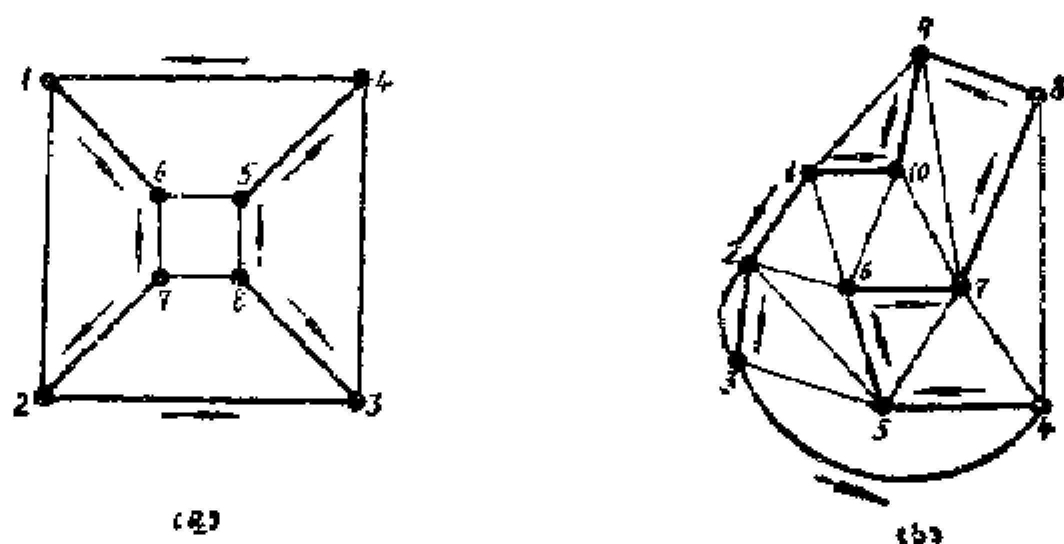
例 2 — 3　图 2 —24中线性图的点 1 和 2 之间所有可能的路径如图 2 —27所示。我们有

$$P_1\oplus P_2=E_6$$
$$P_2\oplus P_3=E_7$$
$$P_3\oplus P_4=E_1$$
$$P_1\oplus P_2\oplus P_3\oplus P_4=E_3$$

式中，E_1，E_6，E_8 和 E_7 已见图 2 —25，可见 $P_1\oplus P_2\oplus P_3$（奇数条路径的环和）就不是一个欧拉图，而是原来给出的线性图，即图 2 —24。

第九节　汉密尔顿路径和回路

一个连通图 G 的欧拉图的特性是 G 的一个闭行走通过每条边确实只有一次。连通图中一个汉密尔顿回路的定义是，它是图中的一个闭行走，因为闭行走是要回到起点的，所以除去起点不算外，其它各点严格地只能通过一次。例如从图 2 —28(a)中的点 1 出发沿着粗线所示的路线，即 1 - 6 - 7 - 2 - 3 - 8 - 5 - 4 - 1 确实通过图中每个点只有一次而又回到起点 1，这就得到了一个汉密尔顿回路。同样，在图（b）中由点 1 出发沿着粗线所示的路线，即 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 -10- 1，通过图中每点确实只有一次而回到起点 1，这就是图（b）中的汉密尔顿回路。更正确地讲，在一个连通图中，如果一个回路包括图中的每一个点，就称为汉密尔顿回路。因此，在有 n 个点的图中的一个汉密尔顿回路确切地包括 n 条边。



(a)　　　　(b)

图　2 —28

很明显，不是每个连通图都有一个汉密尔顿回路的。例如图 2 —23(b)和图 2 —25(g)就没有汉密尔顿回路。这就提出了一个问题，一个连通图能有汉密尔顿回路的必要和充分条件是什么呢？爱尔兰数学家汉密尔顿在 1857 年提出了这个问题，但直到现在还没有很好解决。

如果在汉密尔顿回路中移掉一条边，这就成为一条路径，这条路径有一个特殊的名称，叫做汉密尔顿路径。很明显，一个图中的汉密尔顿路径要通过图中的每个点。既然一条汉密尔顿路径是汉密尔顿回路的一个子图，而汉密尔顿回路又是另一个图的一个子图，那么每个有汉密尔顿回路的图也有一条汉密尔顿路径。可是，反过来讲，很多有汉密尔顿路径的图却

不一定有汉密尔顿回路的。

在考虑有无汉密尔顿回路（或路径）的时候，我们只需要考虑简单图，这是因为一个汉密尔顿回路（或路径）通过每一个点确实只有一次，因此，它不可能包括自环和平行边组的。所以在图中找寻汉密尔顿回路（或路径）之前，一般图可以删除去自环和平行边来先简化它。

什么类型的图能保证有汉密尔顿回路呢？凡有3个或更多点的完全图就能保证。

什么是完全图呢？凡是一个简单图其中每一对点之间都有一条边直接相连的，就叫作完全图。图2—29是有2点，3点，4点和5点的完全图。完全图有时也叫作通用图。既然在一个完全图中，每一个点总有一条边与其它点相连通，所以有n_v个点的完全图，每个点的次数是n_v-1。一个完全图的边数为$n_v(n_v-1)/2$。

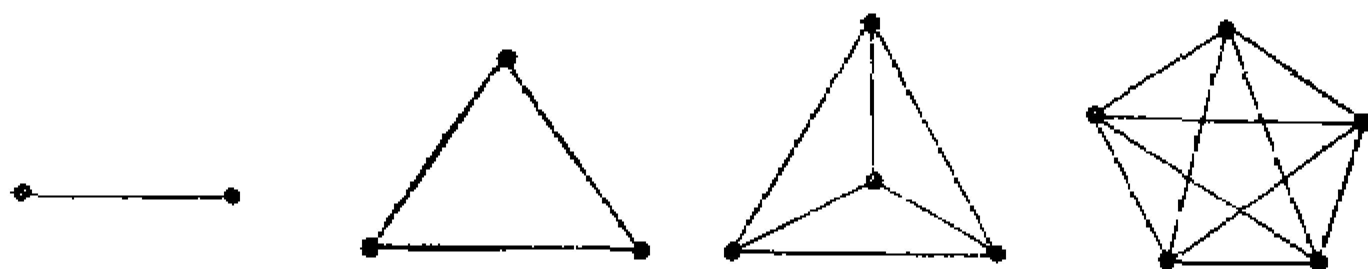


图 2—29

某一给定的图可能包括一个以上的汉密尔顿回路。可是在一个图中，如何决定所有的边不共用的汉密尔顿回路（或路径），一般地讲，也是一个还没有解决的问题。现在只能确定有奇数点$n_v(\geqslant 3)$的完全图中边不共用的汉密尔顿回路数为$(n_v-1)/2$。

第十节 最小运算

由若干个组所构成的一个集体A的最小运算定义如下：

设A是一个集体，那么$\min A$（即A的最小运算符号）是A的子集体，如果A中每个组β，当β不是空组的时候，能满足$\beta\not\subset\alpha$的话，则A中任何组α是在$\min A$之中（其中$\beta\not\subset\alpha$意思是说β不是α的子组，就是说在组β中至少有一个元素是不在α之中）。

例如，假设$A=\{(a,b,c),(a,d),(a),(b,c),\phi\}$。那么$\min A$就包括组$(a)$和$(b,c)$以及空组$\phi$。下列过程会给出一个集体$A$的$\min A$。检验$\alpha_p(\neq\phi)$，其中$p=1,2,\cdots,n$，看是否有一个非空组$\alpha_r(\neq\alpha_p)$以致$\alpha_r\subset\alpha_p$（这里$\alpha_r\subset\alpha_p$是表示$\alpha_r$是一个子组，就是说，$\alpha_r$中某元素是在$\alpha_p$之内），如果这样的组$\alpha_r$，删除$\alpha_p$，否则让$\alpha_p$留在集体中。

例如，图2—24中的线性图可以分解成为图2—25所示7个子图，如再包括空组ϕ在内，则

$\{E\}=\{\phi,(a,b),(c,d),(a,c,e),(b,c,e),(a,d,e),(b,d,e),(a,b,c,d)\}$

应用最小运算，因为组(a,b,c,d)中含有另一非空组(a,b)，它能满足$(a,b)\subset(a,b,c,d)$，所以删除去(a,b,c,d)得

$\min\{E\}=\{\phi,(a,b),(c,d),(a,c,e),(b,c,e),(a,d,e),(b,d,e)\}$

$\{E\}$是所有相异回路，一些回路的边不共用并集，以及空组的集体。设$E_1\subset\{E\}$是回路C_1，C_2，$\cdots C_k$的边不共用并集。那么既然C是一个回路，C_1就是在$\{E\}$之中，而且，$C_1\subset$

E_1，所以通过最小运算后，可以删除E_1以得到$\min\{E\}$，因为$\{E\}$中任何一个回路C是一个连通子图，其中每点的次数为2，那么C的真子组就不会是一个回路。由此得下一个定理。

定理2—7： $$\min\{E\}=\{C\} \qquad (2-14)$$

其中$\{C\}$是所有可能回路和空组的集体。

第十一节　M　图

设E是一个欧拉图，P是点i和点j之间的一条路径，那么$P\oplus E$就是M图。这个环和所得到的M，除点i和j以外的各点都是偶数，而两点i和j却是奇次数。为什么会是这样呢？

设原来有一个图G。现在在G的点i和点j之间增加一条边y以得到一个新图G'。既然y是一条新边，G中的欧拉图E是不包括y的，那么边y必定是在$\{P\cup(y)\}\oplus E$之中。另外很明显，$P\cup(y)$是G'中的一个欧拉图。根据定理2—2既然$\{P\cup(y)\}\oplus E$也是一个欧拉图，那么这个欧拉图中的每个点是偶次数的。从$\{P\cup(y)\}\oplus E$中删除边y后，点i和点j的次数就各减少一，即$d(i)$和$d(j)$都是奇数。其它各点的次数都不受影响。在$\{P\cup(y)\}\oplus E$删除y后，我们得到$P\oplus E=M$。所以M图的定义是：

一个M图是一个线性图，它只有两个点是奇次数的，这两个点称为M图的端点。

图2—30(a)是一个M图，点1和点2是它的端点，次数为奇数，其它各点的次数为偶数。在端点1和2之间增加一条边y后，成为图（b），这样就成为一个欧拉图。从点1开始，我们在图（b）中得一个闭行走（a,b,c,d,e,f,g,y,h,l,m,n），这就包括所有在$M\cup(y)$之中的各边。把这个行走倒转过来，即从点2开始，则得（y,h,l,m,n,a,b,c,d,e,f,g），也是一个闭行走。若去掉边y，从点1开始得到（h,l,m,n,a,b,c,d,e,f,g,h）是一个开行走。它包括M图中所有各边。所以从一个欧拉图中删除一条边就得到一个M图。

图　2—30

一条路径和若干回路的一个边不共用并集是一个边组，它可以分解成一条路径和若干回路的一个边不共用并集，而路径与回路的边不共用并集之间没有共同边。例如在图2—30(a)中，M图中所有边的边组（a,b,c,d,e,f,g,h,l,m,n），可以分解成一条路径（a,b,l）和两个回路的一个边不共用并集（c,d,e,f,g,h,m,n）。因此这个边组是一条路径和两个回路的一个边不共用并集。所以这里得到下面定理。

定理2—8： 一个M图可以是一条路径，或是一条路径和若干回路的一个边不共用并集。

证：从图2—30(b)的举例中，我们知道一个连通图的M图中，有一个包括所有边的开

行走。从本章第七节中我们也知道，如果一个开行走中的所有边不是一条路径的话，我们可以把它分解成为一个开行走和一个闭行走；前者是一条路径，而后者是一个欧拉图。这个欧拉图或是一个回路或是若干回路的一个边不共用并集。因此一个连通图的M图可以是一条路径，或是一条路径和若干回路的一个边不共用并集。

假使一个给定的M图是分离的。设g_1，g_2，…，g_p是M的部分。因为所有点的次数之和的一半是一个线性图的边数，如果有一个奇次数的点是在$g_r(1\leq r\leq p)$之中，那么另一个奇次数的点一定也在g_r之中，因此，只要g_1，g_2，…，g_p之中有一个部分，比如说g_1是一个M图，所有其它部分一定是欧拉图。所以我们可以肯定地说g_1可以是一条路径或是一条路径和若干回路的一个边不共用并集，而每个g_2，…，g_p可以是一个回路，或是若干回路的一个边不共用并集。因此一个给定的M图是一条路径和若干回路的一个边不共用并集。▌

上面在第一章第二节中讲到哥尼斯堡7桥的问题时，我们提到过一笔画出一个图形的问题。如果一个图形能够一笔画出时，它一定是含有图中所有边的一个行走。因此，如果这个行走是一个开行走，这个图一定是一个连通的M图。另一方面，如果是一个闭行走，这个图一定是一个欧拉图。而且，如果不提笔不可能画出一个指定的图形，那么就不能得到一个行走，因此这个图形既不是一个连通的M图，也不是一个欧拉图。这就是说，当且仅当一个图是一个连通的M图或一个欧拉图时，一个图形才可以一笔画出。例如，图2—31(a)所示的线性图是一个连通的M图，所以我们可以一笔画成。可是，在图(b)中的线性图既不是一个M图也不是一个欧拉图，因此我们不能一笔画成。

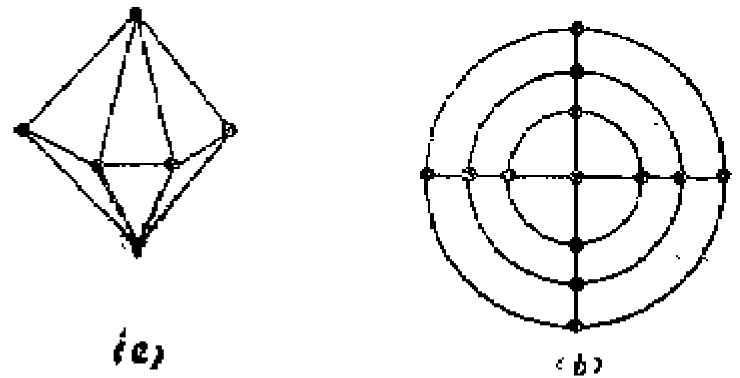


图 2—31

我们已经知道在同一对点之间的两条路径的环和是一个欧拉图。因为一个M图是一条路和若干回路的一个边不共用并集，所以有相同端点的两个M图的环和也将是一个欧拉图。

定理2—9：设一图的两个子图M_1和M_2都是M图，并且端点相同，那么$M_1\oplus M_2$是一个欧拉图。

证：设有M_1和M_2的端点之间另有一条边y，它既不在M_1中，也不在M_2之中。因为$M_1\cup(y)$和$M_2\cup(y)$两者都是欧拉图，所以

$$\{M_1\cup(y)\}\oplus\{M_2\cup(y)\}=M_1\oplus M_2 \qquad (2-15)$$

是个欧拉图。

定理2—10：设一个图中的两个子图之一为一个M图，另一个子图为一个欧拉图E，那么M和E的环和$M\oplus E$是一个M图。

证：根据定理2—8，M可以写成$M=P\oplus E'$，其中P是一条路径，E'是一个欧拉图。因此

$$M\oplus E=\{P\oplus E'\}\oplus E=P\oplus E''=M' \qquad (2-16)$$

其中$E'\oplus E=E''$，根据定理2—2是一个欧拉图。▌

从上面所讲的，我们可以把M图M和欧拉图E的环和关系列成下表2—1。

表2—1与布尔代数中的加法表如表2—2是对称的。

符号$\{P_{ij}\}$表示一个图G中两点i和j间所有可能路径的集体，和符号$\{M_{ij}\}$表示一个图G中以点i和点j为端点的所有可能的子图为M图的集体。符号$\{E\}$的意义前面已经讲

环和运算　　表2—1

⊕	E	M
E	E	M
M	M	E

布尔加法　　表2—2

+	O	1
O	O	1
1	1	O

过，根据定理2－8，有下列关系：

$$\{M_{ij}\}=\{P\oplus E|\ P\in\{P\},\ E\in\{E\}\} \qquad (2—17)$$

现在引进两个术语。设D_1，$D_2\cdots$，D_m是一些组（或集），倘然

$$D_1\oplus D_2\oplus\cdots\oplus D_m=\phi \qquad (2—18)$$

那么D_1，D_2，$\cdots D_m$叫作线性相关组。如果D_1，D_2，$\cdots D_m$之间没有这种关系，那么就叫作线性无关组或线性独立组。

在$\{M_{ij}\}$和$\{E\}$中所有子图的集体$\{E, M_{ij}\}$通过环和运算所得到的是一个阿贝尔群或称交换群。

例2—4　一个集体A包括有$A=\{(a, b), (c, d), (a, d), (b, c)\}$因为

$$(a, b)\oplus(c, d)\oplus(a, d)\oplus(b, c)=\phi$$

所以这4个子组是线性相关的。另一集体B包括有$B=\{(a, c), (b, e), (c, d), (a, d), (b)\}$因为

$$(a, c)\oplus(b, e)\oplus(c, d)\oplus(a, d)\oplus(b)=e\neq\phi$$

所以这5个子组是线性无关的。

定理2—11：在$\{E, M_{ij}\}$中有n_e-n_v+k+1个线性无关组。

例2—5　图2—24中的所有可能欧拉图的集体$\{E\}$和所有可能路径的集体$\{P\}$已见图2—25和图2—27。结果是

$$\{E\}=\{\phi, (a, b), (c, d)(a, c, e), (b, c, e), (b, d, e), (a, d, e)(a, b, c, d)\}$$

$$\{P_{12}\}=\{(e), (a, d), (b, c), (a, c), (b, d)\}$$

所以在端点1和2之间的所有M图为

$$\{M_{12}\}=\{P\oplus E|\ P\in\{P\},\ E\in\{E\}\}$$
$$=\{(e), (a, d), (b, c), (a, c), (b, d), (a, b, e), (c, d, e), (a, b, c, d, e)\}$$

其中首先5个M图是路径，而其余的3个M图是一条路径和若干回路的边不共用并集，后面的3个M图如图2—32所示。

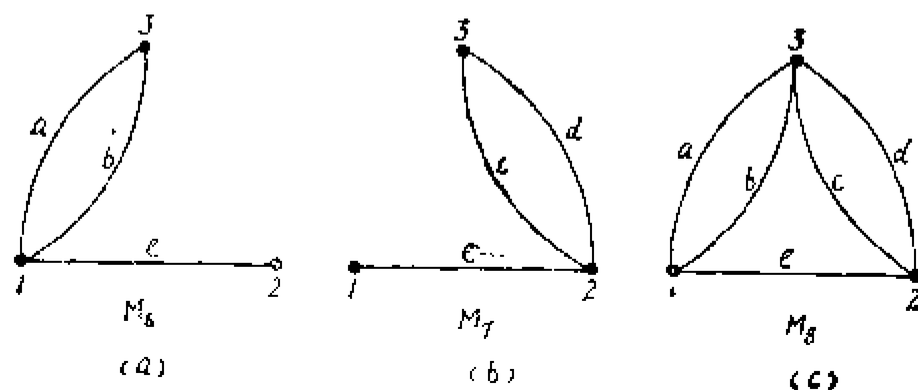


图　2—32

其中$M_6=(a, b, e)$，$M_7=(c, d, e)$，和$M_8=(a, b, c, d, e)$。由线性无关的生成元

组E_1，E_2，E_3，和P_1，通过环和可以产生集体$\{E, M_{12}\}$中所有可能的元素。

$$E_1 \oplus E_2 = E_7$$
$$E_1 \oplus E_3 = E_4$$
$$E_2 \ominus E_3 = E_3$$
$$E_1 \oplus E_2 \oplus E_3 = E_5$$
$$E_1 \oplus P_1 = M_6$$
$$E_2 \oplus P_1 = M_7$$
$$E_3 \oplus P_1 = P_4$$
$$E_1 \oplus E_2 \oplus P_1 = M_5$$
$$E_1 \oplus E_3 \oplus P_1 = P_3$$
$$E_2 \oplus E_3 \oplus P_1 = P_2$$
$$E_1 \oplus E_2 \oplus E_3 \oplus P_1 = P_5$$
$$E_1 \oplus E_1 = E_2 \oplus E_2 = \cdots = P_1 \oplus P_1 = \phi$$

第十二节　路 径 集 体

从上节可知，如果我们适当地选择好生成元，我们可以容易地得到M图的集体$\{M_{ij}\}$。所以点i和j之间得到所有可能路径的一个方法是先在集体$\{E, M_{ij}\}$之中选择生成元来产生集体$\{M_{ij}\}$。然后我们想办法从$\{M_{ij}\}$中抽出那些所有可能路径。上面第二章第十节中所讲的最小运算就是从$\{M_{ij}\}$中抽出所有路径的一种方法。这种运算也可以用来从欧拉图的集体$\{E\}$中抽出所有可能的回路。

通过对$\{M_{ij}\}$的最小运算，我们可以得到点i和点j之间的所有可能路径。所以

$$\min\{M_{ij}\} = \{P_{ij}\} \qquad (2-19)$$

其中$\{P_{ij}\}$是点i和点j之间所有可能集体。

现在再介绍一种环乘积的运算方法，环乘积的符号用$\otimes$。那么什么叫环乘积呢？下面就是它的定义。

集体A和集体B的环乘积，用$\{A\}\otimes\{B\}$来表示，是

$$\{A\}\otimes\{B\} = \min\{\alpha_p \ominus \beta_q \mid \alpha_p \in \{A\},\ \beta_q \in \{B\}\} \qquad (2-20)$$

其中min就是最小运算符号，不过这里运算目的是求一组数据中的最小者或两个数据中的较小者。

例如，设　$\{A\} = \{(a), (b, c), (d, e)\}$和

$\{B\} = \{(a, b), (b, c, d), (e)\}$

则

$$\{A\}\otimes\{B\} = \min\{(b), (a, b, c, d), (a, e), (a, c), (d), (b, c, \theta), (a, b, d, e)\}$$
$$= \{(b), (a, e), (a, c), (d)\}$$

环乘积的一种应用如下，如果在一个不可分图的任何一对点i和j之间有任何一个回路的话，那么在i和j之间就存在着两条路径P_1和P_2，并且

$$P_1 \oplus P_2 = C \qquad (2-21)$$

这就是说，在一个不可分图的任何一对点i和j之间，一个回路可以用两条路径的环和来表

示。那么如果我们在一个不可分图的点 i 和 j 之间所有可能路径中，同时收集每两条路径间环和的话，我们就能得到这个图的所有可能回路。所以有下面这个定理。

定理 2—12：在一个不可分图中

$$\{P_{ij}\}\otimes\{P_{ij}\}=\{C\}。\qquad (2-22)$$

这是因为

$$\{P_{ij}\}\otimes\{P_{ij}\}=\min\{P_r\oplus P_s\,|P_r,\ P_s\in\{P_{ij}\}\}\qquad (2-23)$$

定理 2—13：

$$\{C\}\otimes\{C\}=\{C\}\qquad (2-24)$$

定理 2—14：

$$\{P_{ij}\}\otimes\{C\}=\{C\}\qquad (2-25)$$

这个定理加以改变可得

定理 2—15：

$$\{P_{ij}\}\otimes\{P_{ij}\}\otimes\{P_{ij}\}=\{P_{ij}\}\qquad (2-26)$$

例 2—6　图 2—24中的$\{P_{12}\}$由例 2—5 知为

$$\{P_{12}\}=\{(e),\ (a,\ d),\ (b,\ c),\ (a,\ c),\ (b,\ d)\}$$

为了得到图 2—24中不可分图的所有可能的回路，我们可以应用式（2—21）。根据式（2—22），得

$$\begin{aligned}\{P_{12}\}\otimes\{P_{12}\}=&\min\{(e)\oplus(e),\ (e)\oplus(a,\ d),\ (e)\oplus(b,\ c),\ (e)\oplus(a,\ c),\\&(e)\oplus(b,\ d),\ (a,\ d)\oplus(b,\ c),\ (a,\ d)\oplus(a,\ c),\ (a,\ d)\\&\oplus(b,\ d),\ (b,\ c)\oplus(a,\ c),\ (b,\ c)\oplus(b,\ d),\ (a,\ c)\\&\oplus(b,\ d)\}\\=&\min\{\phi,\ (a,\ d,\ e),\ (b,\ c,\ e),\ (a,\ c,\ e),\ (b,\ d,\ e),\\&(a,\ b,\ c,\ d),\ (c,\ d),\ (a,\ b)\}\\=&\{\phi,\ (a,\ d,\ e),\ (b,\ c,\ e),\ (a,\ c,\ e),\ (b,\ d,\ e),\\&(c,\ d),\ (a,\ b)\}\\=&\{C\}\end{aligned}$$

在应用式 2—22时，应该注意到 (P_{ij}) 的下标 i 和 j，即相同起点和终点的 $\{P_{ij}\}\otimes\{P_{ij}\}$才给出$\{C\}$。如果$\{P_{ij}\}\otimes\{P_{rs}\}$而 i，$j\neq r$、s时，则不能给出$\{C\}$。

定理 2—16：设G为一个连通图，i，j，k为G中 3 个相异点，则

$$\{P_{ij}\}\otimes\{P_{jk}\}=\{P_{ik}\}\qquad (2-27)$$

式中 $\{P_{ij}\}$ 是点 i 和 j 之间所有可能路径的集体，$\{P_{jk}\}$ 是点 j 和 k 之间所有可能路径的集体，$\{P_{ik}\}$是点 i 和 k 之间所有可能路径的集体。

证：设想有一连通图 G，如图 2—33所示，对于$\{P_{ij}\}$中任何一条路径P_{ij}和$\{P_{jk}\}$中任何一条路径P_{jk}，$P_{ij}\oplus P_{jk}$是一个M图M_{ik}，所以我们需要证明的是 $\{P_{ik}\}$ 中任何一条路径 P_{ik}在$\{P_{ij}\}$和$\{P_{jk}\}$中就分别有路径P_{ij}和P_{jk}，使

$$P_{ij}\oplus P_{jk}=P_{ik}\qquad (2-28)$$

既然G是连通的，从点 j 就有一条路径通到路径P_{ik}中任何一点。设P'是从 j 到G中路径P_{ik}中点 r 的一条路径，并且在P_{ik}中不再含有其它点。设P_{ir}是子图 P_{ik} 中点 i 和 r 之间的路径，如图 2—33所示，那么$P_{ij}=P_{ir}\cup P'$，和$P_{jk}=P_{rk}\cup P'$就是所需要的路径。

把这个定理扩大，得下一定理。

定理 2 —17：对于一个连通图G

$$\{P_{ij_1}\}\otimes\{P_{j_1j_2}\}\otimes\{P_{j_2j_3}\}\otimes\cdots\otimes\{P_{j_nk}\}=\{P_{ik}\} \tag{2 —29}$$

其中i，j_1，$j_2\cdots j_n$和k是G的诸点。

同样，根据式（2 —22），我们可以修改上一定理如下：

定理 2 —18：对于一个不可分图

$$\{P_{ij_1}\}\otimes\{P_{j_1j_2}\}\otimes\cdots\otimes\{P_{j_nk}\}\otimes\{P_{ki}\}=\{C\} \tag{2 —30}$$

例 2 — 7 图 2 —24中的P_{23}是$\{P_{23}\}=\{(c),\ (d),\ (a,\ e),\ (b,\ e)\}$

再从例 2 — 5 中所求得的$\{P_{12}\}$，我们可以求$\{P_{13}\}$如下：

$$\{P_{13}\}=\{P_{12}\}\otimes\{P_{23}\}=\{(c),\ (a,\ d),\ (b,\ c),\ (a,\ c),\ (b,\ d)\}$$
$$\otimes\{(c),\ (d),\ (a,\ e),\ (b,\ e)\}$$
$$=\min\{(c,\ e),\ (d,\ e),\ (a),\ (b),\ (a,\ c,\ d),\ (a,\ b,\ d,\ e),\ (b,\ c,\ d),\ (a,\ b,\ c,\ d)\}$$
$$=\{(c,\ e),\ (d,\ e),\ (a),\ (b)\}$$

考虑这个不可分图G'，设e是G'中的一条边，而边e的端点是i和j，如图2 —34所示。设G'_e是由G'删除边e后所得到的图。既然G'_e是连通的，在点i和j之间至少有一条路径。设$\{P_{ij}\}$是G'_e中点i和j之间所有可能路径的集体。注意边e本身就是G'中点i和j之间的一条路径。因此$(e)\oplus P$是一个回路，其中P是在$\{P_{ij}\}$之中。换句话说，在$\{P_{ij}\}$中任何P，在G'中就存在着一个回路，使

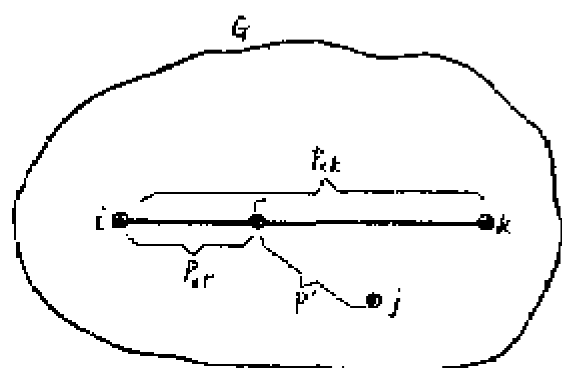


图 2 —33

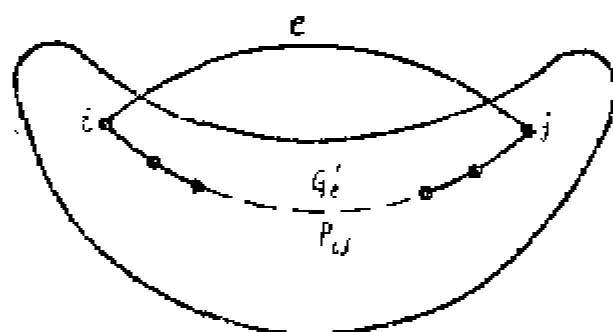


图 2 —34

$$C\oplus(e)=P \tag{2 —31}$$

设$\{C\}$是G'中包括空组和所有可能回路的集体。既然$(e)\oplus\phi=(e)$，集体的最小运算$\min\{(e)\oplus C\ |C\in\{C\}\}$是点$i$和$j$之间所有可能路径的集体。

符号$\{P_{(e)}\}$是边e的端点之间所有可能路径的集体。

定理 2 —19：对于一个图中任何一条边e

$$\{P_{(e)}\}=\min\{(e)\oplus C|\ C\in\{C\}\} \tag{2 —32}$$

证：设$P_1(\neq\phi)$是点i和j之间的一条路径。那么$P_1\cup(e)$是$\{C\}$中的一个回路。空组ϕ是在$\{C\}$之中。因此

$$\{(e)\oplus C\ |C\in\{C\}\}\supseteq\{P_{(e)}\} \tag{2 —33}$$

$(e)\oplus C$是一个M图，它的端点就是边e的端点。因此

$$\min\{(e)\oplus C|\ C\in\{C\}\}=\{P_{(e)}\} \tag{2 —34}$$

例2 — 8　例2 — 6 中不可分图中的$\{C\}$是$\{C\}=\{\phi,\ (a,\ b),\ (c,\ d),\ (a,\ c,\ e),\ (b,\ c,\ e),\ (b,\ d,\ c),\ (a,\ d,\ e)\}$由式（2 —32）

$$\{F_{(a)}\}=\min\{(a)\oplus C\mid C\in\{C\}\}$$
$$=\min\{(a),\ (b),\ (a,\ c,\ d),\ (c,\ e),\ (a,\ b,\ c,\ e),\ (a,\ b,\ d,\ e),\ (d,\ e)\}$$
$$=\{(a),\ (b),\ (c,\ e),\ (d,\ e)\}$$

这个结果正是例2—7中所求得的$\{P_{13}\}$。所以

$$\{P_{12}\}\otimes\{P'_{23}\}\otimes\{P_{13}\}=\{P_{13}\}\otimes\{P_{13}\}=\{C\}$$

定理2—20：设$\{E\}$是一个图G的所有欧拉图和空组的集体。假设P是$\{P_{ij}\}$中的一条路径，那么

$$\{P_{ij}\}=\min\{P\oplus E\mid E\in\{E\}\} \qquad (2—35)$$

证：设P'是$\{P_{ij}\}$中任何一条路径，那么，$P\oplus P'$是一个欧拉图，所以$P\oplus P'$是在$\{E\}$之中。

路径集体的一种应用是在开关理论方面，我们可以把线性图作为一个开关网络来考虑，其中每条边代表着一个开关变数，那么从i到j的开关函数F_{ij}可以表示为

$$F_{ij}=\sum_{q}P_{q_{ij}}\text{乘积} \qquad (2—36)$$

其中$\sum_{(q)}$表示从i到j的所有可能q条路径的和，“$P_{q_{ij}}$乘积”表示在路径$P_{q_{ij}}$中与若干条边有关的所有开关变数的乘积。所以如果我们知道点i和j之间所有可能的路径，我们就可以得到开关函数F_{ij}。现在举例来说明这种方法。

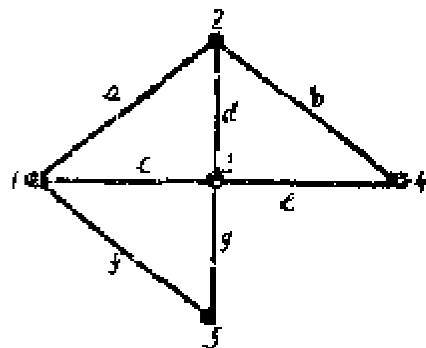


图 2—35

例2—9 图2—35是一个线性图，可以作为一个开关网络来考虑，其中a，b，c，d，e，f，g是开关变数。假设我们需要找出开关函数F_{12}和F_{14}。当然，我们可以从网络中直接求得点1和2之中间所有可能的路径和点1和4之间所有可能的路径。可是，我们也可以如下应用式（2—14），（2—27），（2—32）和（2—35）来求得。首先，我们先求$\{E\}$

$$\{E\}=\{\phi,\ (a,\ c,\ d),\ (c,\ f,\ g),\ (b,\ d,\ e),\ (a,\ d,\ f,\ g),\ (a,\ b,\ c,\ e),\ (b,\ c,\ d,\ e,\ f,\ g)(a,\ b,\ e,\ f,\ g)\}$$

那么$\{c\}=\min\{E\}=\{\phi,\ (a,\ c,\ d),\ (c,\ f,\ g),\ (b,\ d,\ e),\ (a,\ d,\ f,\ g),\ (a,\ b,\ c,\ e),\ (a,\ b,\ e,\ f,\ g)\}$

因为 $\{P_{12}\}=\{P_{(a)}\}$，我们得到

$$\{P_{12}\}=\{P_{(a)}\}=\{(a),\ (c,\ d),\ (d,\ f,\ g),\ (b,\ c,\ e),\ (b,\ e,\ f,\ g)\}$$

因此，

$$F_{12}=(a+cd+dfg+bce+befg)$$

因为$\{P_{24}\}=\{P(b)\}$，

$$\{P_{24}\}=\min\{(b)\oplus C\mid C\in\{C\}\}=\{(b),\ (d,\ e),\ (a,\ c,\ e),\ (a,\ e,\ f,\ g)\}$$

根据式（2—27）

$$\{P_{14}\}=\{P_{12}\}\otimes\{P_{24}\}=\{(a),\ (c,\ d),\ (d,\ f,\ g),\ (b,\ c,e),\ (b,\ e,\ f,\ g)\}\otimes\{(b),\ (d,\ e),\ (a,\ c,\ e),\ (a,\ e,\ f,\ g)\}$$

$=\{(a, b), (a, d, e), (c, e), (e, f, g), (b, c, d), (b, d, f, g)\}$

即 $$F_{14}=ab+ade+ce+efg+bcd+bdfg$$

因为 (a, b) 是点 1 和 4 之间的一条路径，根据式（2—35）我们可以得到$\{P_{14}\}$为

$$\{P_{14}\}=\{P(a, b)\}=\min\{(a, b)\oplus E \mid E\in\{E\}\}$$
$$=\{(a, b), (b, c, d), (a, d, e), (b, d, f, g), (c, e), (e, f, g)\}$$

这个结果是与应用式（2—27）所得的结果是完全一样的。

第十三节 巡回检查工作的旅程问题

假使有一位领导同志想要到他所管辖的范围内去巡回检查工作，他要去的几个地方是知道的，以及各地之间的距离也是知道的。他应该如何计划他的旅程，使得他能够检查到每个地方的工作，而每个地方只经过一次，最后回到他工作的地方呢？

这个旅程问题外国叫推销员问题，也有译作货郎担问题，而与汉密尔顿回路有密切联系的。我们可以把所要到的地方用点来表示，而两地之间的铁路或公路用边来表示，这样就得到了一个线性图。既然两地之间的距离是已知的，所以每条边e_i上可注以一个正实数R，这个数就是两地之间的里程。这个数有时叫作权。边上注以数的图叫作加权图。在我们的问题中，如果两地之间都有直接的铁路或公路可通，那么这个旅程所代表的图叫作完全加权图。这种图中有很多汉密尔顿回路，我们的问题就变成从其中找出总里程（或权）最短的那个汉密尔顿回路。

在一个有n_v点的完全图中，不同汉密尔顿回路（当然不是边不共用的）的总数是$(n_v-1)!/2$。这个数字是这样推导出来的，即从任一点出发；我们可以从这第一点有(n_v-1)条边可选择，余类推，这些选择都是互不相关的，所以可能的选择总数为$(n_v-1)!$。因为在这种选择中，每个汉密尔顿回路算了两次，所以$(n_v-1)!$应除以 2，即得上述结果。

从理论上讲，巡回检查工作的里程问题只要把所有$(n_v-1)!/2$个汉密尔顿回路的各个回路的里程总数算出来，然后挑选其中最短的。可是实际上，如果n_v数很大，这个计算工作量就很可观了。比如说当n_v为50时，即使用数字计算机来计算、工作量也是太大的！

所以当n_v数大时，问题在于规定一个易于处理的算法来寻找最短的里程，到目前为止（指1974年）n数为任意大小的里程问题虽然有很多人在研究，但是还没有找到一个有效的算法。这个问题在运筹学中有很多用处。有几个特殊的例子已经研究出来。现在也有几个直接推断解法，可以给出很接近于最短里程的旅程，但是都不能保证最短的。

对于点数n_v少的有向图有一种算法可以找出汉密尔顿路径，将在第十一章第八节中介绍。

第十四节 邮递员问题

邮递员所管的投递区每天要去送信，每次每条街巷至少要经过一次，其中有些街道要经过一次以上。每条街巷都有长度，采取怎样的路径，从每点出发，每条街巷至少经过一次，而总行程为最短呢？我国数学家管梅谷教授于1962年提出了一篇论文解决了这个问题，国外图论著作中称为中国邮递员问题。

如果邮递员所管辖的投递区画成线性图时是一个欧拉图，那么因为一个欧拉图可以从一

点出发经过图中每条边只走一次回到原出发点，这个问题就不成为问题了。但是一般的投递区画成线性图时总会有些点的次数是奇数的，那么如何解决最短的路径呢？

假设投递区的线性图中有奇次数的点，根据定理 1 — 1，一个图中奇次数的点数一定是偶数的。设这个数为 $2n$，这 $2n$ 个点可以连成某种形式的 n 个行走。在这 n 个行走中有些边 E' 必须经过两次的，那些经过两遍的 n 个行走将成为 n 个回路。这些 n 个回路连同那些偶次点将规定一个方法来得到一个回路，其中每边至少经过一次。我们以其经过 E' 边两次，简单些的办法还不如在这图中增加平行于 E' 的一套边。这样所得的图，每边只要经过一次，为了简单明了计，这些新增的平行边称为新增边。

加入了新增边后所得到的图是一个欧拉图，因此就有了一个闭行走，即图的欧拉线。邮递员所走的总行程是原图各边的长度和再加上新增边的长度。原图各边的总长度是给定的，现在的问题是如何增加最小数的新增边了。所以管教授的邮递员问题可以说成是这样的一个问题，就是给定一个连通图 G，每边有一长度，图中共有奇次点数为 $2n$，找出连通 $2n$ 个点的几条行走，使包括 n 条行走的各边总长度为最小。

例如有一个投递区如图 2 —36所示，其中奇次点有 4 个，即点 1 、 2 、 4 和 8 ，这 4 个奇次点之间可以组成不同方案的两个行走，图 2 —37就是方案之一，其中虚线是新增的。增加了新增边后，各点的次数都是偶数，所以这图是一个欧拉图，可以得到一条欧拉线。很明显，解法是很多的，但是为了使加了新增边后邮递员的总行程最短的一个最优解法，管教授提出了下一定理，规定了两个优化条件：

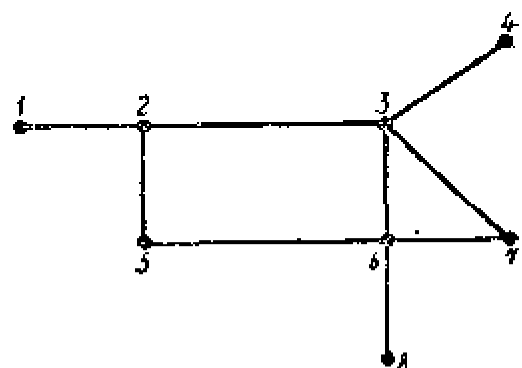


图 2 —36

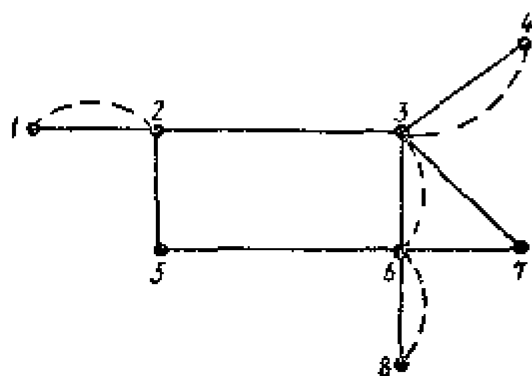


图 2 —37

定理 2 —21：对于邮递员问题优化解法的充分条件为：

（一）无冗余边；

（二）对于每一回路中新增边的总长度不超过这个回路总长度的一半。

这个定理中的第一个条件所谓冗余边是指平行于原图中任何边的两条或更多的新增边。结果一个可行的解法与第一个条件相矛盾，那么可以删除新增边中的两条冗余边。所得结果仍旧是一个可行的解法，而新增边的总长度较少，这就表示先前的方案不是优化的。这可以图 2 —38为例。如在点 1 、 2 、 4 、 8 之间选择由点 1 至点 4 和点 2 至点 8 之间的两个行走，分别如图 2 —38中点划线和虚线所示，这个方案中就在点 2 和点 3 之间出现了冗余边。删除这两点之间的冗余边后就如图 2 —37所示。

如果所提的解法与定理 2 —21的第二个条件相矛盾，那么至少有一个回路中新增边的总长度将大于这个回路总长度的一半。很明显这个回路中有几条边是原图中的边，而另有几条边是新增边。在这种情况时，就有可能取消回路中的新增边，而代之以回路中属于原图中较短边的平行边，结果就得到一个可行的回路而总长度较小。这就表示原来的解法不是优化的。例如，在图 2 —39（a）中点 1 和点 6 是奇次点，这就需要在点 1 和点 6 之间加入新增

边以便组成一个行走成为一个回路。假使新增边如图（b）所示。从这里可以看出其中有两条新增边是属于一个回路的，这就是点 2 与点 4 之间的边和点 4 与点 5 之间的边。这个回路如图（c）所示。这个回路的长度是$6+8+5+4=23$。新增边的长度和是14，它大于这个回路长度的一半，所以与定理 2 —21的第二个条件相矛盾。在这种情况时，我们可以删除属于这个回路中的新增边（即图（c）中的虚线）而代之以这个回路属于原图的两条边的平行边作为新增边，结果将如图（d）所示。注意，在图（d）中新增边的长度和是小于图（b）中新增边的长度和的。

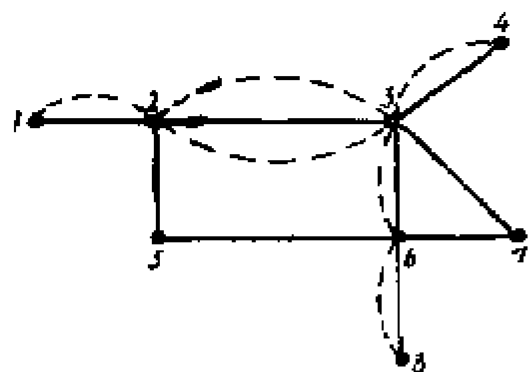


图 2 —38

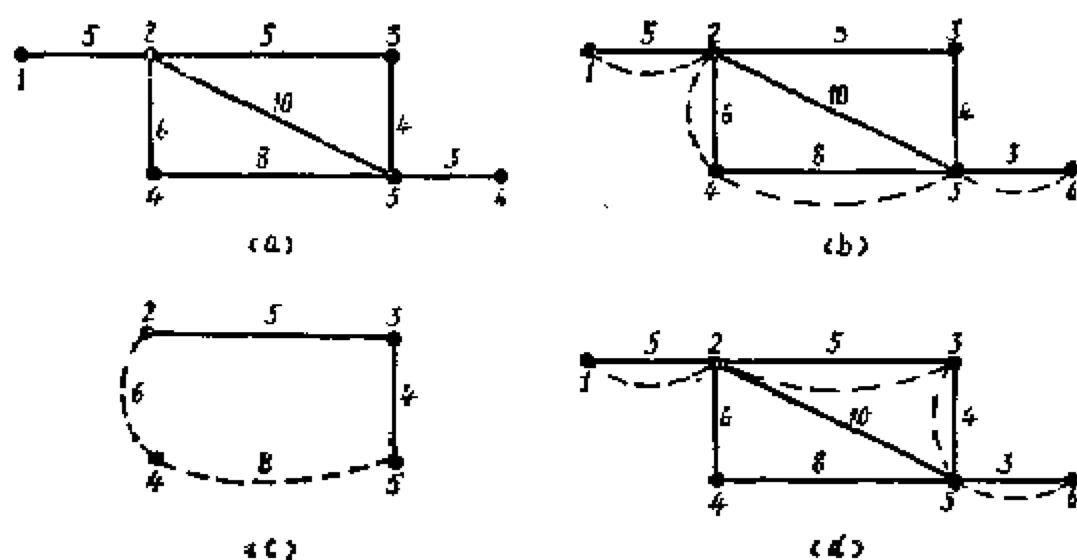


图 2 —39

习　题

2 — 1　证明图 2 —13中的图（a），（b），（c）是同构的，并标出对应的点和边。

2 — 2　证明图 2 —40中的两个图是同构的。

图 2 —40

2 — 3　证明如果一个连通图G分解为两个子图g_1和g_2，则在g_1和g_2间一定至少存在一个共用点。

2－4　证明在一个连通图G中移去一条边e_i时，当且仅当这条边e_i存在于G中的一个回路时，剩下的图是连通的。

2—5　列出图2—41中点5和点6间所有不同路径，并绘出每条路径的段数。

2—6　在一个图G中，设P_1和P_2是两个给出点间的两条不同路径。证明$P_1 \oplus P_2$是G中的一个回路，或是G中的一个回路的边不共用并集。

2—7　画出一个图，要求在这个图中的一条欧拉线也是一个汉密尔顿回路。对于这样的图，你能说明些什么？

2—8　证明如果具有n_v个点的图G中每一对点v_i、v_j的次数满足下列条件

$$d(v_i)+d(v_j)\geqslant n_v-1$$

时，则图G总是有一条汉密尔顿路径。

2—9　试证明具有两个或两个以上点的简单图，至少有两个点的次数相等。

2—10　求出图2—12中全部欧拉图的集体$\{E\}$，并用定理2—20求取$\{P_{13}\}$。

2—11　在图2—43（a）和（b）中，你能分别找到含有所有边的行走吗？写出这些行走。

2—12　有一邮递员的投递区如图2—44所示，每边所注的数字为街道的长度，试问该邮递员自点1出发送信时应采取怎样的路径？

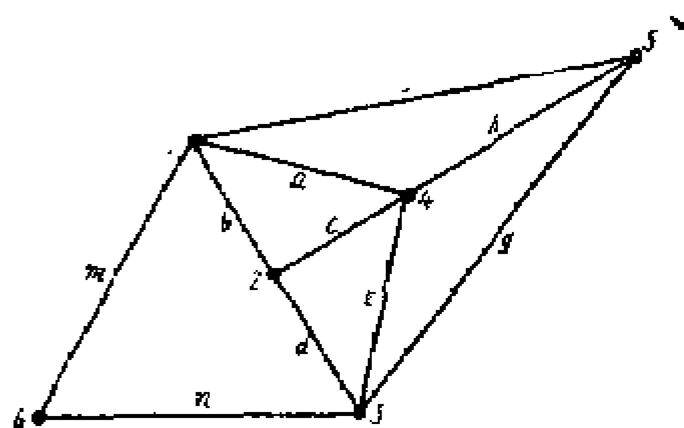


图　2—41

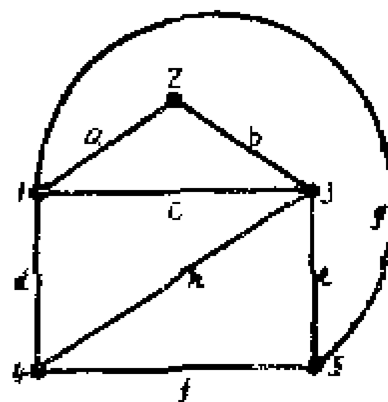
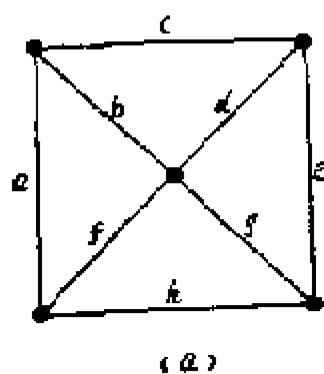


图　2—42



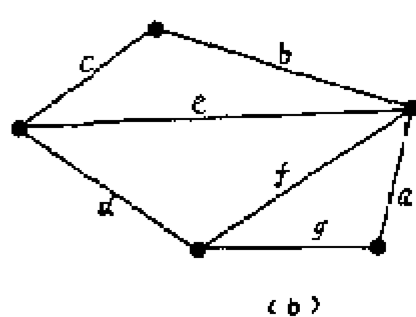


图　2—43

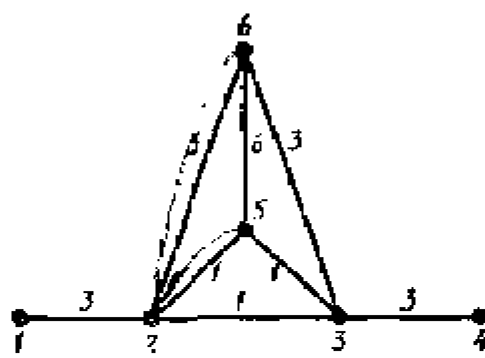


图　2—44

第三章　树

第一节　树的定义和特性

树是一种没有回路的连通线性图。树的概念是图论中最重要的概念之一。

在第一章中已经讲过，一个图至少要有一个点。树也是这样，至少有一个点。图3—1是有一点、两点、三点和四点的树。图3—1（e）也称为星图，它的中心点与周围的点直接相连。当然，一般的树有更多的点。有的书中称没有点的树为零树，但是我们以为既然连点也没有，就不成其为树了。同样，我们讲的是有限图，所以以后所讲的树也是有限的。

既然树是一种没有回路的连通图，所以是一种简单图。要是它有自环或平行边，这两者都能组成回路，这就与树的定义相矛盾了。

在日常的事务关系中，树的情况是很多的，封建时代的家谱或族谱就是用树来代表的。河流的支流和分支也可以用树来表示。邮政传递的编码和穿孔卡片的分类也都是按照树的形式来编制的。后者的树叫作分类树或判定树。

现以我国准备采用的邮政编码为例。这是一种四级六码制。我国的邮政投递系统分四级，即省（市、自治区），邮区，县和投递区。我国的邮政编码用六位，前两位数代表省、市或自治区；第三位数代表邮区（一个省可以划分为几个邮区）；第四位数代表县；第五位，第六位两位数代表投递区。如北京市的邮政编码首两位数为10，代表北京市大邮区；第三位数0代表市区，1代表京东邮区，2代表京西邮区；第四位数代表县；最后两位数代表投递区；如图3—2所示。

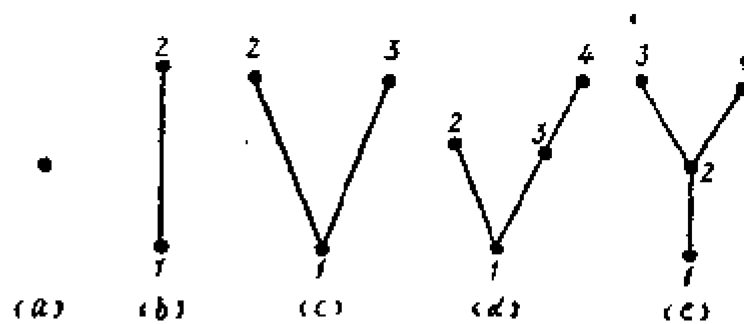


图　3—1

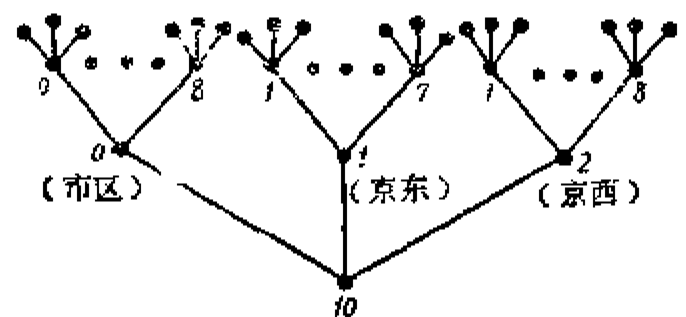


图　3—2

在很多分类问题中，我们在每一个中间点只能有两种选择，譬如说大或小，好或坏，对或不对，0或1，这就是二分法。这样在每一个点只有两种选择的判定树常常出现在计算机程序设计和开关理论中。这将在下面讨论到。

现在我们先研究树的一些特性。

定理3—1：在一棵树T中，每一对点之间只有一条路径。

证：因为T是一个连通图，在T中每一对点之间一定至少有一条路径。现在假使在T的两点a和b之间有两条不同的路径，那么这两条路径的并集将含有一个回路。这样T就不是一棵树了。▮

相反地：

定理 3 — 2： 如果在一个图G中每一对点之间只有一条路径，这个图G是一棵树。

证：在一个图G的每一对点之间存在着一条路径保证着G是连通的。在一个有两个点以上的图中的一个回路暗示着至少有一对点a和b之间有两条不同的路径。因为G的每一对点之间只有一条路径，G就不能有回路。所以G是一棵树。

定理 3 — 3： 有n_v个点的一棵树有n_v-1条边。

证：这个定理可以根据点数用归纳法来证明。对于$n_v=1$，2和3的树(参看图3—1)，这个定理的正确性是容易知道的。假设这个定理对于少于n_v个点的所有树是对的。

现在设一棵树有n_v个点。在T中，设边e_k的端点为i和j。根据定理3—1，在点i和j之间，除了e_k外没有其它路径。所以从T中删除e_k将使图分离如图3—3所示。而且，$T \ominus e_k$正好包括两个部分，并且因为T中没有回路，所以每个部分是一棵树。这两棵树，T_1和T_2中每棵树的点数都少于n_v。那么根据归纳假设，每棵树中的边数比点数少一。因此$T \ominus e_k$包括n_v-2条边（n_v个点）。所以T正好有n_v-1条边。▮

定理 3 — 4： 每一个有n_v个点和n_v-1条边的连通图是一棵树。

我们已经注意到一棵树的另一个重要特征是它的点是由最少的边数连接起来的。如果一个连通图中的任何一条边移去后使图成为分离的，这个图就叫作是最少连通的。一个最少连通图不会有回路的，否则我们即使从回路中移去其中的一条边，这个图还仍旧是连通的，因此一个最少连通图是一棵树。相反地，如果一个连通图G不是最少连通的，在G中一定存在着一条边e_i，$G \ominus e_i$还是连通的，因此e_i是在某一个回路之中，这就意味着G不是一棵树。所以有下一个定理。

定理 3 — 5： 当且仅当一个图是最少连通的，它是一棵树。

这个定理的重要性是很明显的。我们可以直觉地知道把n个不同的地点连接起来，所需要的最小线段是$n-1$。即使没有电工基础知识的人也会了解到，要把n个接线柱短接起来，至少需要$n-1$条导线。根据定理3—5，这个结构就是一棵树。

我们已经知道，有n_v个点而没有任何回路的一个连通图有n_v-1条边。我们也可以从下一定理看出有n_v个点而没有回路和有n_v-1条边的图总是连通的，就是说它是一棵树。

定理 3 — 6： 有n_v个点，n_v-1条边，并且没有回路的一个图G是连通的。

证：假使一个无回路的图G有n_v个点和n_v-1条边，而是分离的话，那么G将包括两个或更多的无回路的部分。无损于普遍性，设G包括两个部分g_1和g_2，如图3—4所示，我们在g_1中的点1和g_2中的点2之间加入一条边e。因为g_1和g_2是分离的，即点1和2之间没有路径，即使加入一条边e也不会产生一个回路。因此$G \cup e$是一个有n_v个点和n_v条边的无回路的连通图。根据定理3—3，这是不可能的。

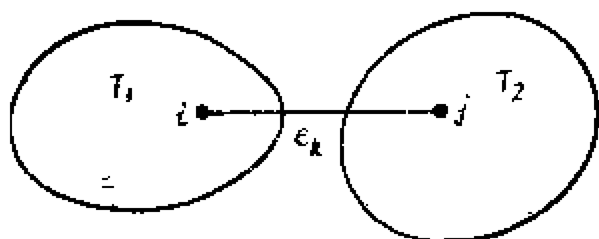


图 3—3

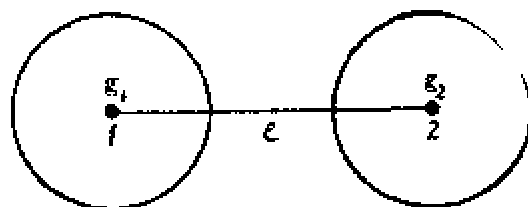


图 3—4

上述六个定理可以归纳成下列五个不同的但是等效的树的定义。就是说，如果具有n_v个点的一个图G有下列五个条件之一的叫作一棵树：

（一）G是连通的，并且是无回路的；

（二）G是连通的，并且有n_v-1条边；

（三）G是无回路的，并且有n_v-1条边；

（四）在G中每一对点之间只有一条路径的；

（五）G是一个最少连通图。

从图3－1中可知：每棵树（一点的树除外）有几个悬挂点，而悬挂点的次数为一。树的悬挂点有时称为树叶。

定理3－7：一棵树T中，至少有两个点的次数为1。

证：从式（1－7）中我们知道一个图的各点的次数的和为边数的两倍。一棵树的边数为n_v-1，其中n_v为树的点数。所以树的所有点的次数之和为

$$\sum_{i=1}^{n_v} d(v_i)=2(n_v-1) \qquad (3-1)$$

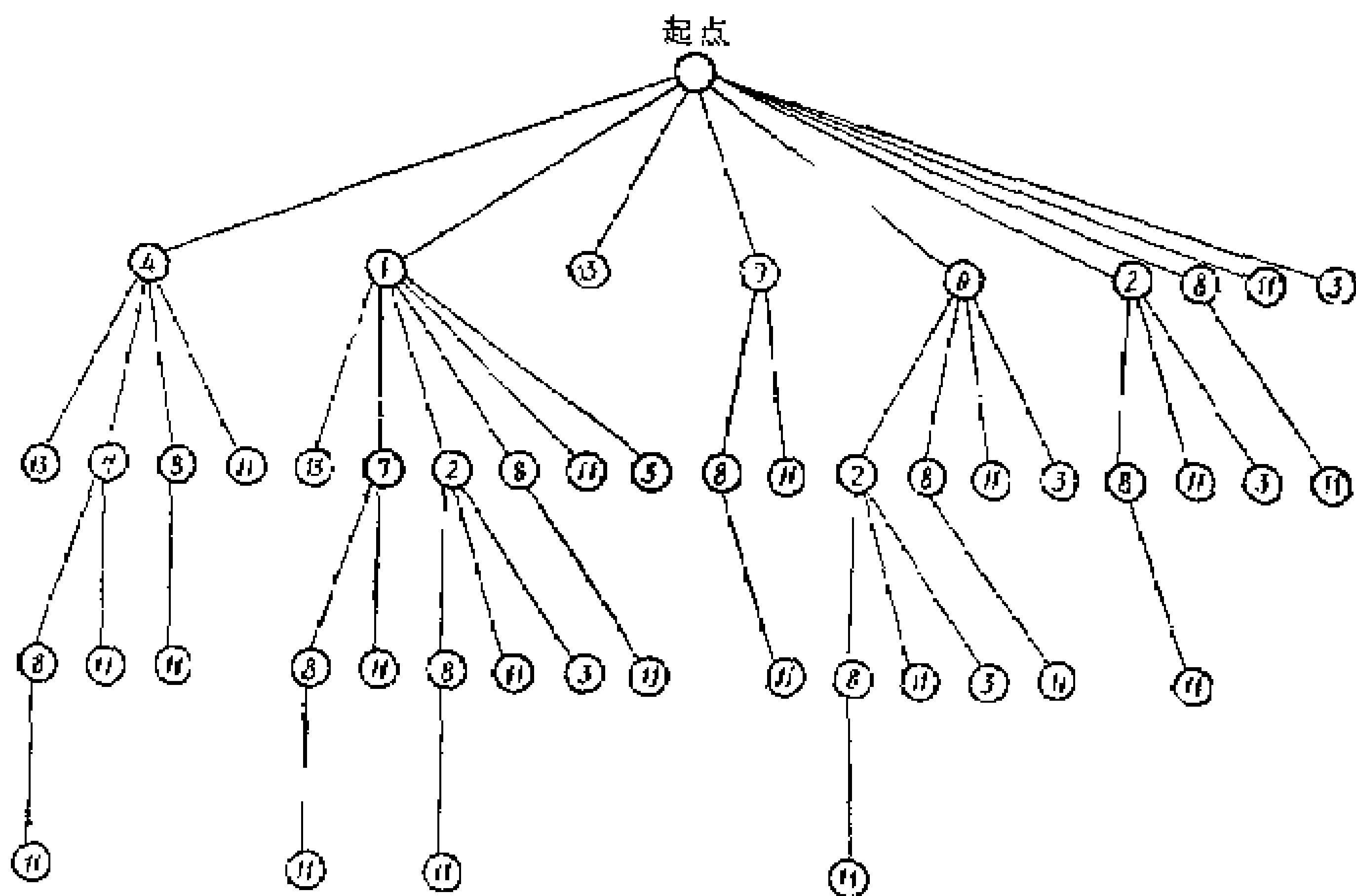
假设树的各点的次数等于或大于2，那么

$$\sum_{i=1}^{n_v} d(v_i)\geqslant 2n_v \qquad (3-2)$$

但是式（3－1）明确地指出$\sum_{i=1}^{n_v} d(v_i)$不可能大于$2(n_v-1)=2n_v-2$。这就说明一定有几点的次数小于2。假使树中只有一个点的次数为1，而其它各点的次数都至少为2，那么

$$\sum_{i=1}^{n_v} d(v_i)\geqslant 2(n_v-1)+1 \qquad (3-3)$$

这个结果还是大于$2n_v-2$，因为点的次数不会是零的，这就证明了一棵树中至少有两个点的次数为1。



定理3－－7：也可以写成是：在任何一棵树T中，至少有两个悬挂点。

在树中，两点之间的间隔称为段。在加权图中，就称为长度。在树中，有一点与其它各点不同，如图3－1中的点1，这点叫作根，也叫参考点。

图论广泛地应用于计算机的程序设计中。在讲到程序设计时，有时可能提出这样一个问题，譬如给出一个整数序列，其中各数都是不同的，要求找出其中的最大单调递增子序列。假使给出的序列是4，1，13，7，0，2，8，11，3。解法可用树来表示，其中除始点外，各点都代表序列中各别数字。由始点到各个点i的路径就描述终点i的单调递增子序列，如图3－5所示，就是（4，7，8，11），（1，7，8，11），（1，2，8，11）和（0，2，8，11）。每个子序列的段数都是4。这样一种用来代表数据的树，计算机程序设计员称之为数据树。

第二节 树的距离和中心

在图3－6中的树有4个点。直觉地看起来，似乎点2比其它3点更处于中心。在讨论这个问题之前，我们先要对距离下一定义。

在一个连通图G中，点组中两点i和j之间的距离$d(i,j)$是它们之间最短路径的段数，即最短路径的边数。

关于两点之间的距离的定义，非但对树而言，也适用于其它任何连通图。在一个不是树的图中，在两点之间一般有几条路径。我们必须列举所有路径，并且找出其中最短一条路径的段数（有时可能有几条最短路径）作为两点之间的距离。

例如图3－7中，在点1和2之间的一部分路径为（a，e），（a，c，f），（b，c，e），（b，f），（b，g，h）和（b，g，l，m）等。其中最短的路径有两条，即（a，e）和（b，f），每条的距离为两段。所以$d(1,2)=2$。

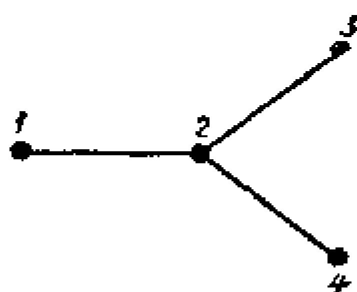


图 3－6

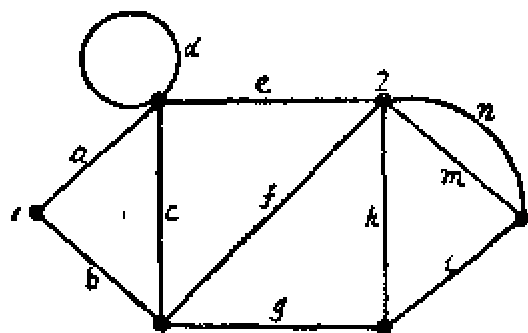


图 3－7

在树中，既然任何两点之间只有一条路径，所以距离的决定要容易得多。例如，在图3－6的树中，$d(1,2)=1$，$d(1,3)=2$，$d(1,4)=2$，$d(3,4)=2$，$d(2,3)=1$，和$d(2,4)=1$。

现在再回到我们在本章开头所讨论的问题，先学习一个术语。

在一个图G中，一个点i的偏心率$E(i)$是从这个点i到图G中距i最远的那个点k的距离，即

$$E(i)=\max_{k\in G} d(i,k) \qquad (3-4)$$

式中max d（　）表示最远距离。

偏心率也有人称为连带数或隔离度。

图G中有最小偏心率的那个点叫作图G的中心。图3－6中4个点的偏心率为$E(1)=2$，$E(2)=1$，$E(3)=2$，和$E(4)=2$。所以点2是这棵树的中心。再看图3－8，图中各点旁的数字表示该点的偏心率。这棵树有两个点的偏心率都是同样的小。所以这棵树有两个中心。

一般地讲，一个图（不一定是树）可以有很多中心。例如成为一个回路的多边形，每个点就是一个中心。可是对树而讲，那么有下列定理。

定理3－8：每棵树有一个或两个中心。

证：从一棵树的某一点i到另一点j的距离$d(i,j)$，只有当j是一个悬挂点时才是最大。一棵有两个以上点的树T，它就至少有两个悬挂点（定理3－7）。先从T上删除各悬挂点，这仍是一棵树T'。T'各点的偏心率是怎样的呢？稍一考虑就会了解到从T上删除各悬挂点后，就是把T上余留下来各点的偏心率普遍地都减少1。因此T中的中心点还是T'中的中心点。从T'再删除各悬挂点后得T''。如此继续进行下去，如图3－9（a），（b），（c），（d）所示，最后只剩一个点或一条边。一个点时，这个点就是树的中心；一条边时，这条边的端点就是树的两个中心。这就证明了这个定理。

从定理3－8的论证中可以得到一个推论，就是，如果一棵树T有两个中心，这两个中心一定是相邻的。

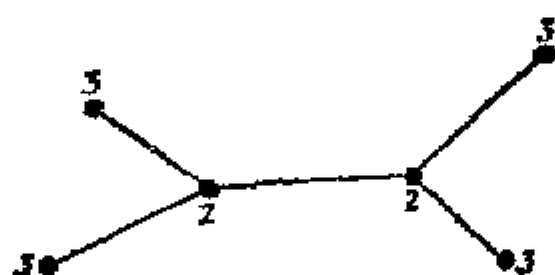
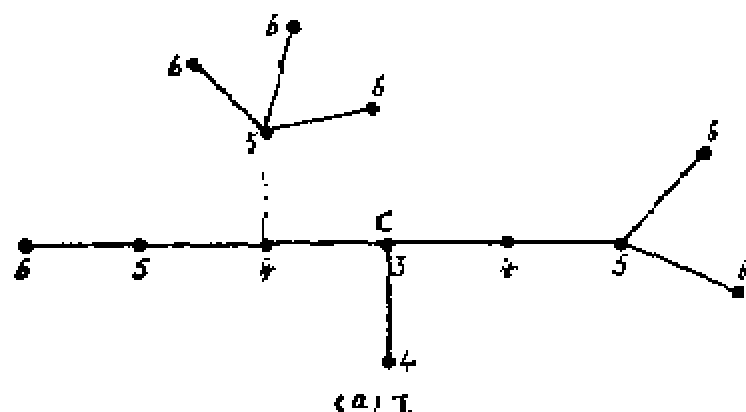
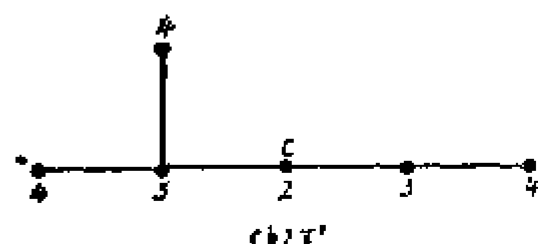


图 3－8





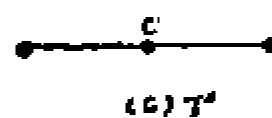


图 3－9

图论的应用范围很广，非但在自然科学方面用到，就是在社会科学方面也有用处。如果有一个由13人组成的团体，用图3－9（a）来表示，其中点代表团体的成员，而边则代表他们之间的通信联络线。因为线性图是连通的，所以这个团体的所有成员都是可以互相联系的，或者直接联系，或者通过别的成员来联系。但是这里也应注意到，这个图是一棵树，连通是最少的。所以这个团体不能失去任何一条联络线。每点的偏心率$E(i)$代表一个成员与团体中最远成员之间的紧密程度。图3－9（a）中点c的位置应该是这个团体的负责人的位置。这样才能体现一个负责人与团体中其它成员紧密联系的关系。

如果一棵树有一个中心（或者两个中心）的话，那么是否也有半径呢？是的，树是有半径的。在一棵树中，中心的偏心率就是树的半径。例如，在图3－9（a）中，树T的半径是3或3段。树T的直径是树中最长的路径。例如，图3－9（a）中T的直径是6或6

段。这里有一点是值得注意的，就是在图论中，树的半径不一定是它的直径的一半。这从图3—8中就可以看出。

第三节 有根树和二元树

树根的定义在本章第一节中已经讲过，凡是有根的树就称为有根树。在图3—1中，点1就是根。在图3—5中，起点也可以看作是树根。关于有根树将在第七章的第五节中再讲。一般地讲，“树”是指没有根的。为了区别于有根树，没有根的树有时叫作自由树或无根树。

有一种有根树叫作二元有根树。因为这种二元有根树可以广泛地应用来研究计算机的检索法，二进制识别问题和可变长度二进制码，所以特别值得注意。二元树的定义就是一种树，其中只有一个点的次数为2，而其它各点的次数为1或3（当然，很明显，我们这里讲的是有3个或更多点的树），如图3—10所示。既然次数为2的那个点与众不同，这个点就作为根。所以二元树是一种有根树。二元树有两个特性：

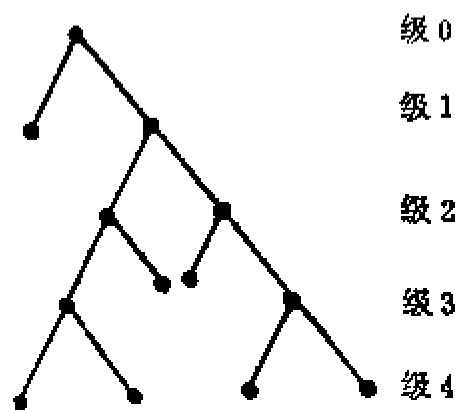


图 3—10

（一）二元树的点数n_v总是奇数的。这是因为树中只有一点的次数为偶数，而其它n_v-1个点的次数为奇数。由定理1—1可知一个图中奇次数的点数总是偶数，即n_v-1是偶数，那么n_v就是奇数了。

（二）设n_p是二元树T的悬挂点数，那么n_v-n_p-1是次数为3的点数。所以T中的悬挂点数为

$$\frac{1}{2}\left[n_p+3(n_v-n_p-1)+2\right]=n_v-1$$

即

$$n_p=\frac{n_v+1}{2} \qquad (3-5)$$

树中的非悬挂点叫作内部点。一棵二元树的内部点数比悬挂点数少1。在一棵二元树中，一个点i距离根的段数l_i就是这个点i的级别或级l_i。所以根是在级0。如图3—5所示的13个点，4个级的二元树在级1，2，3和4的点数分别为2，2，4和4。

二元树的最简单应用法之一是在于检索过程。二元树中每一个点代表着有两个可能出路的检测。我们先从根开始，在根处检测的结果把检索送到下一级的两个点去。在那里再进行检测。就这样继续进行下去，等待到达检索目标的那个特定悬挂点，检索就终止。像这样一种检索过程，重要的是建立这样一棵二元树。如有给定的点数n_v，就应该把离根最远的那个点尽可能地接近那个根。很明显，只有树根那一点是级零，级1的点至多为两点，级2的点至多为四点，余类推。所以k级二元树最多可能的点数为

$$2^0+2^1+2^2+\cdots+2^k\geqslant n_v \qquad (3-6)$$

在一棵二元树中，最大级l_{max}叫作树的高度。一个有n_v个点的二元树的最低可能高度为

$$\min l_{max}=\lceil\log_2(n_v+1)-1\rceil \qquad (3-7)$$

式中$\lceil n\rceil$表示大于或等于n的最小整数。树的高度是以段数计的，所以一定是个整数。

另一方面，建立一个有n_v个点的二元树而使最远点离根越远越好的话，除了级零外，每级必须只能有两个点。所以最远点的高度为

$$\max l_{max}=\frac{n_v-1}{2} \qquad (3-8)$$

对于$n_v=11$的二元树，为实现有最低可能高度，二元树的结构应如图3—11所示。

由式（3—7）得

$$\begin{aligned}\min l_{max}&=\lceil \log_2(11+1)-1\rceil\\&=\lceil \log_2 12-1\rceil\\&=\lceil 3.58-1\rceil\\&=\lceil 2.58\rceil\\&=3\end{aligned}$$

为了实现有最高可能高度二元树的结构应如图3—12所示。

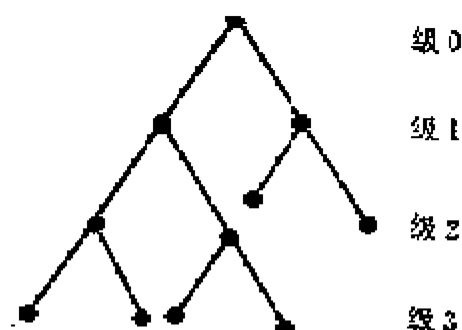


图 3—11

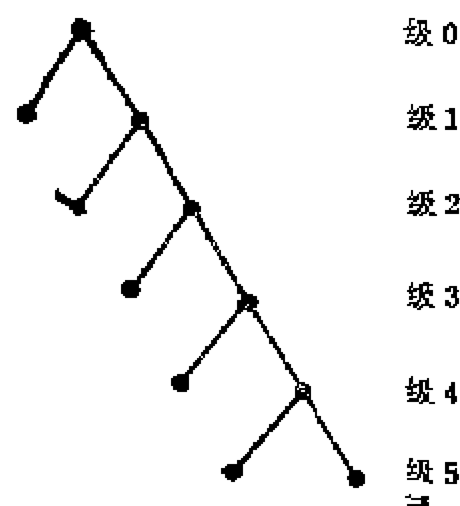


图 3—12

由式（3—8）得

$$\max l_{max}=\frac{11-1}{2}=5$$

第四节　标　记　树

当一棵树的各点都标以不同的名称或标记，这种树就叫做标记树。一个图的各点都标以不同的名称或标记时，这种图就叫作标记图。当我们计算不同图的数目时，有无标记的区别是很重要的。

图3—13示出有4个标记点的16棵树。从这里可以看出，计算图数时，有无标记是大不相同的。图3—13中第一行的4个图的结构是一样的，所以是同构，但是只因为它们各点的标记不同，所以算作4棵不同的树。如果4点都不加标记，那末这四棵树只能算作一棵。如果对图3—13细心检查，有四个点的无标记树，即A，B，C，D四点间无区别的话，只能算作两棵。

下面介绍英国数学家凯莱提出计算标记树的定理，即著名的凯莱定理。

定理3—9：有n_v个点（$n_v\geqslant 2$）的标记树的数目是$(n_v)^{n_v-2}$。

关于这个定理曾有许多人从不同角度作过证明，都是比较复杂的，这里就从略了。在图3—13中，有4个标记点，即$n_v=4$，所以标记树数为$(n_v)^{n_v-2}=(4)^{4-2}=16$。

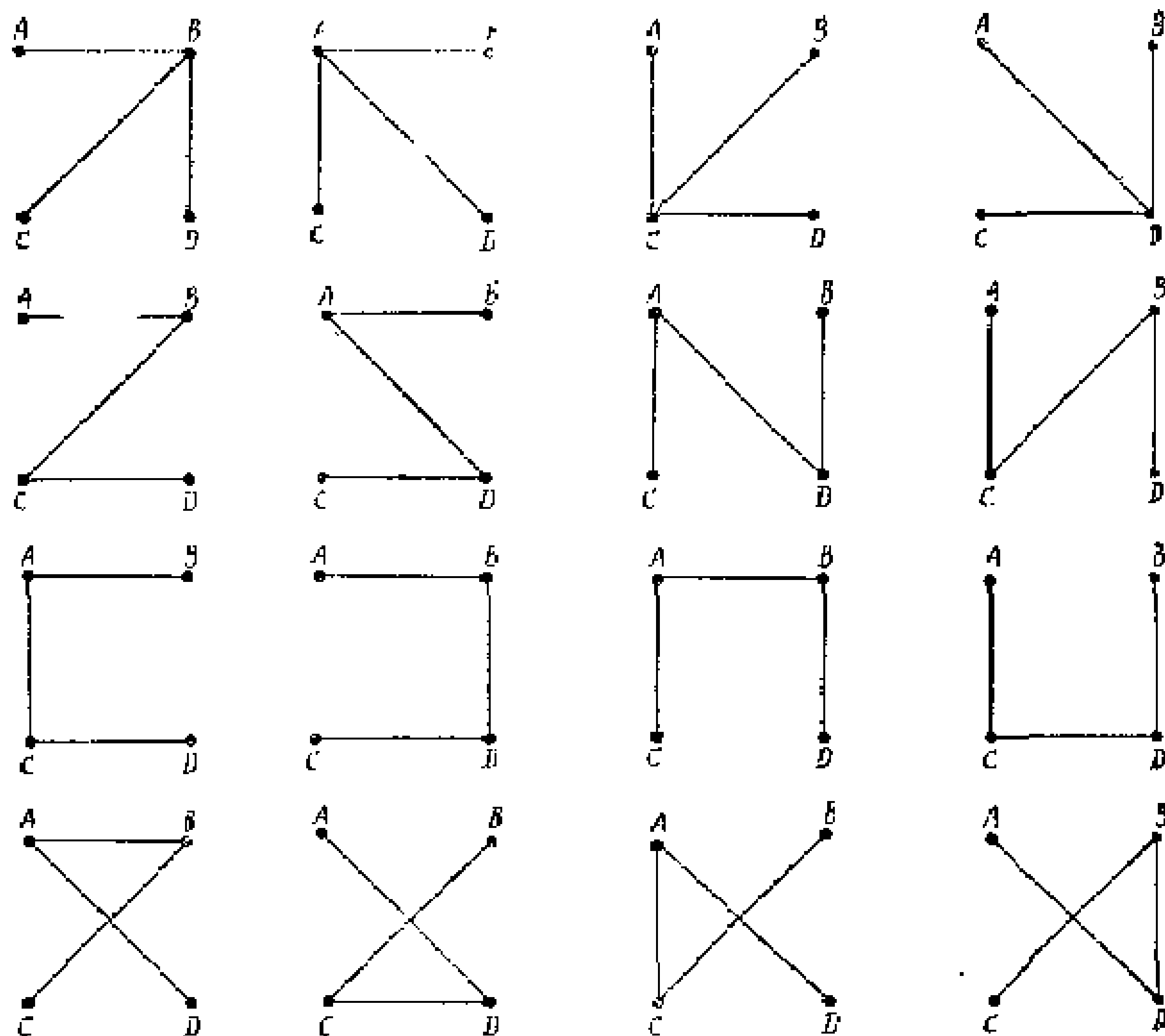


图 3－13

第五节 生 成 树

上面所讨论到的树都是图的本身就是树形。现在我们将研究那些作为另一图的子图的树。一个有n_e条边的图可能有2^{n_e}个不同组合的子图，显然其中有些子图是树。

如果一棵树T为一个连通图G的子图，并且包括G中所有的点，这树就称为生成树。例如，在图3－14中，用粗线表示的子图就是这个图G的生成树。由于图G中的各点都一起挂在生成树上，所以生成树有人认为是G的骨架。既然生成树是G中子图为树中的最大的树（即包括最多的边数），所以也可以叫它是G的最大树子图或最大树。

因为树总是相连的，所以只有连通图才有生成树。在一个有n_v个点的分离图中，找不到有n_v个点的连通子图的。可是，分离图的每一个部分（部分是连通的）是有生成树的。所以有k个部分的分离图有一个生成林，后者由k个生成树所组成，若干树的集体称为林。

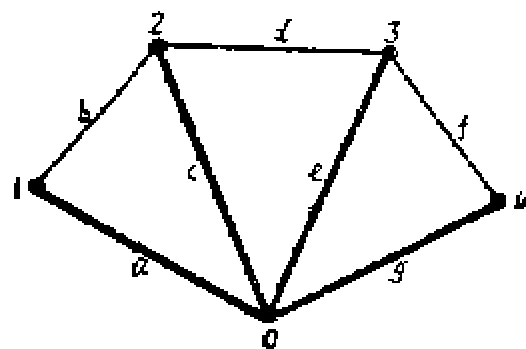


图 3－14

从一个连通图G中找寻一棵生成树是很简单的。如果G中没有回路，那么G本身就是一棵生成树。如果G有一个回路，从回路中删除一条边，这仍旧保持G是连通的。如果G有更多的回路，那么从回路中删除一条边的工作重复下去，直到从最后的一个回路中删除一条边．使G仍旧是连通的，没有回路的，但是保留着各点的图。这样我们有下一个定理。

定理 3－10：每一个连通图至少有一棵生成树。

每一棵生成树T的一条边叫作T的树枝，或简称枝。G中不包括在那棵生成树T中的那些边叫作弦。在电工原理中，有时把弦叫做连支。例如，在图3－14中，粗线的边a，c，

e 和 g 是生成树的枝，而细线的边 b，d 和 f 是生成树的弦。这里有一点应该注意，就是枝和弦是指某一棵生成树而言的。在一个图 G 中，某条边是一棵生成树 T_1 的枝，却可能是另一棵生成树 T_2 的弦。

有时，可以把一个连通图 G 看作是一棵生成树 T 和这棵树的余图 $\overline{T}$ 的并集，即

$$T \cup \overline{T} = G \tag{3-9}$$

既然子图 $\overline{T}$ 是弦的集体，所以很合适地把它看作是 T 的弦组或连支组或余树。由生成树的定义可以得到下一定理。

定理 3—11：对某一生成树而言，一个有 n_v 个点和 n_e 条边的连通图有 n_v-1 个树枝和 n_e-n_v+1 条弦。

例如，图 3—14有 7 条边（$n_e=7$）和 5 个点（$n_v=5$），对生成树（a，c，e，g）而言，它有 $5-1=4$ 个树枝 a，c，e 和 g，以及 $7-5+1=3$ 条弦 b，d 和 f。任何其它一棵生成树将有同样多的树枝和弦。

假使我们有一个电网络，它有 n_e 个元件（相当于边）和 n_v 个节点（相当于点）。那么我们至少要移去几个元件才能使这网络没有回路呢？答案是 n_e-n_v+1。同样，如果有如图 3—15所示那样一块土地，由田埂隔成 6 块田，田中都灌着水。现在要问至少要挖开几条田埂才能把水排出？这里可以把图 3—15看作是一个连通图，其中有点 $n_v=10$ 和边 $n_e=15$。现在的问题是移去几条弦时可使这图成为一棵生成树？由定理 3—11知道这个连通图变成生成树时，应该只有 9 个枝（$10-1=9$）和 6 条弦（$15-10+1=6$）。我们只要适当地选择把 6 条田埂挖开，就能把这块土地的水排出。

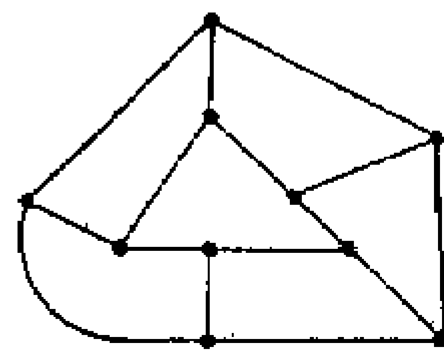


图 3—15

第六节 图的主要数据

当人们要说明一个图 G 时，首先他最可能是提到 G 的点数 n_v，接着就是 G 的边数 n_e，然后是 G 的部分数 k。如果 G 是一个连通图，$k=1$，要是分离图的话，$k\geqslant 2$。一个图 G 中，n_v，n_e 和 k 这三者的关系是怎样的呢？既然一个图的每个部分至少要有一个点，所以 $n_v\geqslant k$。而且，在一个部分中的边数不可能比这个部分的点数减一为少，即 $n_e\geqslant n_v-k$。不管上述的这两个约束（即 $n_v-k\geqslant 0$ 和 $n_e-n_v+k\geqslant 0$），这三个数 n_v，n_e 和 k 是独立无关的，并且是线性图的基本数据。不过也应该指出，单独这三个数字一般是不足以详细说明一个图的。

从这三个数字可以推导出两个重要名称：即秩数 R 和圈数 N。

$$R=n_v-k \tag{3-10}$$

$$N=n_e-n_v+k=n_e-R \tag{3-11}$$

一个连通图的秩数 R 为 n_v-1，它的圈数 N 为 n_e-n_v-1。例如图 3—14的秩数为 4，圈数为 3。关于图 G 的秩数和圈数可以注意有下列关系：

图的秩数 R = 图的任何一棵生成树的树枝数 （3—12）

图的圈数 N = 图的任何一棵生成树的弦数 （3—13）

秩数 + 圈数 = 图中的边数 （3—14）

第七节　基 本 回 路

因为一棵树的任何两点之间已经存在着一条路径，所以在它们之间加一条边，就是增加了一条路径，就形成了一个回路。下面这一定理就不难理解了。

定理 3 —12：当且仅当在一个连通图G的任何两点之间增加一条边时正好产生一个回路的，这个图G是一棵树。

试考虑一个连通图G中的一棵生成树T。加任何一条弦到T就正好产生一个回路。这种由增加一条弦到一棵生成树面形成的回路称为一个基本回路。

一个图有多少基本回路呢？它的数目正好是弦数或圈数N（$=n_e-n_v+K$）。那么一个图总共有多少回路呢？我们知道增加一条弦到一棵树产生一个回路。如果我们再加上一条弦后，是否只多产生一个回路呢？如果在树上同时增加所有的弦后又将发生什么呢？

如以图 3 —16为例，粗线为生成树（b_1，b_2，b_3，b_4，b_5，b_6）。加一条弦c_1后，得一个子图（b_1，b_2，b_3，b_4，b_5，b_6，c_1），它有一个基本回路（b_1，b_2，b_3，b_5，c_1）。如果我们加到树上的是弦c_2面不是c_1，我们会得到一个不同的基本回路（b_2，b_3，b_5，c_2）。现在假设我们把c_1和c_2两条弦都加到那棵树上，子图（b_1，b_2，b_3，b_4，b_5，b_6，c_1，c_2）非但得到我们刚才提到的基本回路，而且还得另一个不是基本回路的回路（b_1，c_1，c_2）。在图 3 —16中，虽则共有75个回路（系由计算机计算出来的），不过其中只有 8 个是基本回路，每一个是粗线所示的生成树加一条弦。

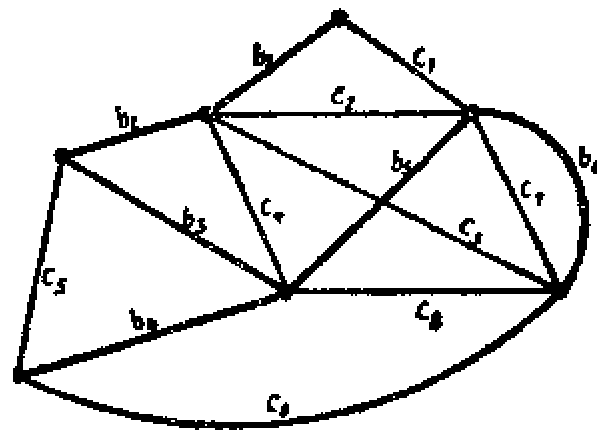


图　3 —16

这里有两点要加以说明。第一，一个回路是否是基本回路系对某一棵生成树面言的。某一个回路对一棵生成树面言可能是基本回路，但是对这同一图的另一棵生成树面言，就不一定是了。虽然一个图的基本回路数（以及回路总数）是固定的，但一个回路却视有关生成树而变成基本回路或否。

第二，大多数应用中，我们所感兴趣的不是一个图的所有回路，而只是一组基本回路，而后者是容易找出来的。基本回路的概念是由克希荷夫采用的，在电网络的分析中有重大意义。克希荷夫指出过不管一个网络有多少回路，我们只需考虑对任何一棵生成树面言的基本回路，面其它回路只是若干基本回路的组合。

第八节　生成树的产生和变换

在一个给定的连通图中，通常是有很多生成树的。在应用中，我们需要所有的生成树。那么，怎样去产生生成树呢？一个合理的方法是先在图中找出一棵生成树，譬如说T_1，如图 3 —17（b）中的（a，b，c，d）。在T_1中加一条弦，譬如h，这就成为一个基本回路（b，c，h，d）。从这个基本回路中移去一个枝，譬如c，这就产生一棵新的生成树T_2，如图 3 —17（c）所示。这种在一棵生成树中通过加一条弦并删除另一合适的枝来产生另一棵新的生成树的办法叫作循环交换，也叫做初级树变换。这里顺便提一下，为了解

决某种运输问题，这种变换是迭代序列中的一种标准运算方法。

在上述步骤中，我们可能不是删除枝 c 而是删除 d 或 b，这样我们会得到另外两棵生成树（a，b，c，h）和（a，c，d，h）。产生这三棵生成树后，每棵中是有弦 h 的，我们可以再从 T_1 开始，加进一条不同的弦，如 e，f 或 g，重复上述过程，从所形成的基本回路中每次删除一个枝以得到一个不同的生成树。这样我们就有了一个从任何一个给定图中产生生成树的方法。

在线性电网络的拓扑分析中，基本上就变成在相应的线性图中如何产生树。所以找寻一个有效过程来产生一个图的所有的树是一个很重要的实际问题。

上述过程也提出很多疑问。靠若干次循环交换，我们能否从任何一棵生成树来产生出一棵所需要的生成树呢？照这种方式我们能否产生某一个图的所有生成树呢？这种交换枝弦的过程要继续进行多长时间呢？从我们可以开始交换的所有可能生成树中，是否有一棵值得优先考虑开始的生成树呢？在这一章中，我们只能试图回答其中某些问题，其它一些问题可能在以后的章节中找到，而有些也许要找文献来研究。

一个图 G 的两棵生成树 T_i 和 T_j 之间的距离是指 G 中有些边在一棵树中有而在另一棵树中没有的边数。这个距离可以用符号 d（T_i，T_j）来表示。在图3—17中，d（T_2，T_3）$= 3$。

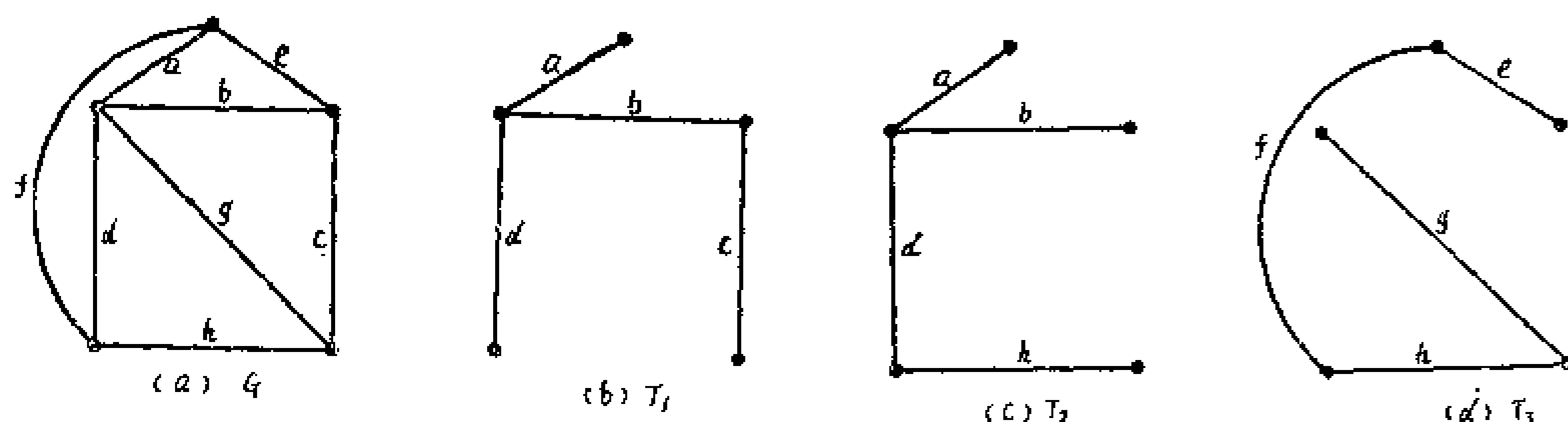


图 3—17

这一个图 G 的两棵生成树 T_i 和 T_j 的环和为 $T_i \oplus T_j$，另设 n_e（g）表示图g中的边数，由树间距离的定义知

$$d(T_i, T_j) = \frac{1}{2} n_e(T_i \oplus T_j) \qquad (3-15)$$

式中距离 d（T_i，T_j）不难看出就是一个图中由 T_i 变换到 T_j 时的最少循环交换数。在生成树的产生和变换中，有下列定理。

定理3—13：由一个图 G 的任何一棵生成树开始，靠连续的循环交换法，我们可以得到 G 的每一棵生成树。

既然在一个秩数为 R（即图的点数为 $R+1$）的连通图 G 中，一棵生成树有 R 条边，我们就有下列结果：

G 中任何两棵生成树间的最大距离为

$$\max\ d(T_i,\ T_j) = \frac{1}{2} \max\ n_e(T_i \oplus T_j) \leqslant R \text{（}G\text{的秩数）}$$

$$(3-16)$$

同样，如果N是G的圈数，我们知道，为了得到另一棵生成树T_j，在T_i中所替换的边数不多于N。所以

$$\max\ d(T_i,T_j)\leqslant N \qquad (3-17)$$

合并式（3－16）和（3－17）得

$$\max\ d(T_i,T_j)\leqslant \min(N、R) \qquad (3-18)$$

其中min（N，R）是图G中N和R这两个数字中的较小者。

第九节　最短生成树

在第二章末讲过，当一个图的每条边注以权时，这个图就叫作加权图。一个图G的一棵生成树T的权是T中各枝的权之和。一般地讲，G中不同的生成树将有不同的权。G的所有生成树中，权最小的那棵生成树是有实际意义的。在加权图中，有最小权的那棵生成树叫作最短生成树。

最短生成树的一个可能应用是在交通网络方面。假使我们要把几个城市v_1，v_2，…，v_n通过交通网联系起来，有些城市之间可以修建直达公路，也有些城市之间不能修建直达公路。在有条件造公路的两个城市v_i和v_j之间建造一条直达公路的费用是可以知道的。现在的问题就是找出把这几个城市联系起来的最少费用的公路网。很明显的，因为我们总是可以在一个图中移去几条边而得到一个有较低权的连通图，所以这个联系网络是一个树形。因此用最低费用建造连接n个城市的公路网问题就是在有n个点的加权连通图中找出一棵最短生成树的问题。一棵最短生成树的必要和充分条件是由下一定理決定的。

定理3－14：当且仅当一个加权连通图G的一棵生成树T没有另一棵与T的距离为1的其它生成树的权小于T的权时，T是最短生成树。

证：这个定理告诉我们最短生成树T的必要条件是明显的，否则我们可以用一次循环交换来得到另一棵比T为较短的生成树了。其实，这个条件也是非常充分的，但是不够明显。所以我们作如下的证明。

设T_1是G中的一棵生成树满足这个定理的假设，即没有一棵与T_1的距离为1的其它生成树的权比T_1为较小。如果我们能够证明G中的T_2不同于T_1，但T_2是一棵最短生成树，它的权与T_1的权相等，证明就算完成。设T_2是一棵最短生成树。因此T_2一定也满足定理的假设，否则又会有另一棵与T_2的距离为1的其它生成树比T_2为较小。这当然与假定T_2为最短相矛盾的。

取T_2中的一条边e，它不在T_1之中，将边e加到T_1与T_1中的诸枝形成一个基本回路，这些与边e形成基本回路的T_1中的枝，虽然不是全部，但是可能有一些也在T_2之中。这些T_1中的那些枝的权，由于对T_1所作的假设，它们的权或小于e或等于e的权。如果所有这些枝的权都小于e的权，那么又与对T_2所作的假设相矛盾了。因此，在T_1中至少有另一个枝（系与e形成基本回路中的）的权与e的权相同。通过一次循环交换，用边e来代替这个枝，产生另一棵生成树T_1'，它的权与T_1的相同，不过T_1'多了一条与T_2共同的边，并且T_1'也满足定理3－14的条件。这种论证可以重复进行，产生着一系列等权树T_1，T_1'，T_1''，……每棵树与T_2的距离为1，直到最后得到T_2自己。这证明与T相距为1的生成树没有一棵是比T为短的。这就是说T是最短的生成树。▮

为了实际地找寻一个图的最短生成树，现在可采用的算法有几种，关于计算方法，有的

用手工，有的用计算机。这里介绍两种算法。

有一种算法是先把图G的所有各边照递增权的次序列成一表。其次选出G中最短的边。然后每下一步从G中所有留下边中选出与前次选出的诸边不成为回路的另一条最短边。这样继续下去一直等到$n-1$条边已经选出为止。这些选出的边就组成这个图的最短生成树。这个算法的有效性就是根据于定理3—14的。现举例来说明这个算法。

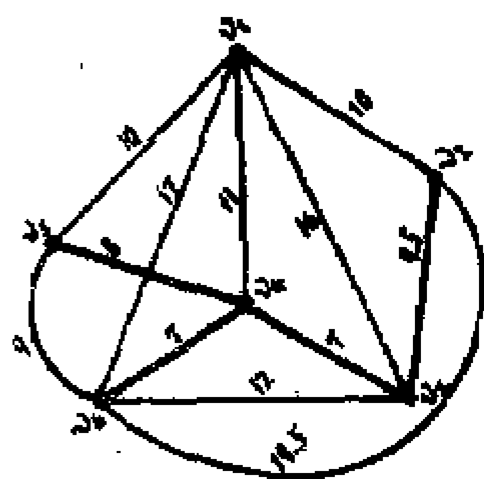


图 3—18

设有一个有6个点和12条边的图如图3—18所示。先把6个点标以名称，如v_1，v_2，…，v_6，把每两点之间的已知权注在那两点之间的边上。然后按照递增次序把12条边的实数列表如下：

边名	权	选出	边名	权	选出
(v_3,v_4)	7	✓	(v_1,v_5)	10	
(v_4,v_6)	7	✓	(v_1,v_4)	11	
(v_4,v_5)	8	✓	(v_3,v_6)	12	
(v_5,v_6)	9		(v_1,v_3)	16	
(v_2,v_3)	9.5	✓	(v_1,v_6)	17	
(v_1,v_2)	10	✓	(v_2,v_6)	19.5	

有了这个表后，选最短边就很容易了。先选出边(v_3,v_4)，把这一行的实数7加一个✓表示选中，并顺手再在图3—18上把边(v_3,v_4)用色笔画粗。照同样方法再选边(v_4,v_6)和(v_3,v_5)，并把表3—1中的7和8两个数分别加一个✓表示选中，再用色笔把所选的边画粗。再选下去的话，照表3—1中的次序，似乎应该是边(v_5,v_6)了，因为那条边的权9为留下诸边中最短的了。可是如果选出这条边后，它就将与边(v_4,v_6)和(v_4,v_5)组成一个回路。这是不允许的。所以这条边不能选用而只能选边(v_2,v_3)，再下一步选边时，边(v_1,v_2)和(v_1,v_5)两者中只能选一条而不能都选，因为都选用后，原来已经选用的边(v_3,v_4)，(v_4,v_5)，(v_2,v_3)，与边(v_1,v_2)，(v_1,v_5)又将形成一个回路。这是不允许的。所以两者中只能选用一条边。现在我们就用边(v_1,v_2)。如果我们这次选用了边(v_1,v_5)，我们会得到另外一棵有同等权的最短生成树。到此为止，图3—18中的6个点已经可以用5（$=6-1$）条边连接起来。这就是这个图的最短生成树，它的权是41.5。

另一种算法的列表方法是与前一种算法有所不同的。现仍举上例来说明这种方法。

先将6个点列成6×6表格形式，再把v_i和v_j之间所给出的实数值填入对应空格中。

	v_1	v_2	v_3	v_4	v_5	v_6
v_1	−	10	16	11	10	17
v_2	10	−	9.5	∞	∞	19.5
v_3	16	9.5	−	7	∞	12
v_4	11	∞	7	−	8	7
v_5	10	∞	∞	8	−	9
v_6	17	19.5	12	7	9	−

各值排列的位置对称于对角线，而对角线是空的。图3—19中不存在的边（相当于两个城市之间没有直达公路可以建造的）的权作为无穷大（∞）。由点v_1开始，在行1中找出最小的表值。由表可知，这是（v_1，v_2）或（v_1，v_5）。设我们采用（v_1，v_5）。如果我们在这里采用（v_1，v_2）的话，我们会得到另外一棵与图3—18中粗线所示的有同等权的最短生成树。现在我们考虑（v_1，v_5）作为一个子图，即把点v_1和v_5用色笔或粗线连接起来。次从子图（v_1，v_5）连接到最近的邻点（即从行1和行5中找出除v_1和v_5以外最小的表值。由表可见这是v_4。在点v_5和v_4之间画一色笔或粗线，并把v_1，v_5，v_4所连成的树作为一个子图。继续照上述方法找出另外三个最小的值（v_4，v_6），（v_4，v_3）和（v_3，v_2），把图3—19中的6个点用（6—1）条边连接起来。这在图3—19中用粗线表示出来，这就是这个图的最短生成树。这棵最短生成树虽然与用前一方法所得的结果不一样，但是这棵树的权也是41.5，与图3—18粗线所示的权是相同的。由此可见，一个图的最短生成树可能有几棵的。从建造公路网的观点看，图3—18和图3—19中粗线所示这两个方案的建设费用是同样最低的。

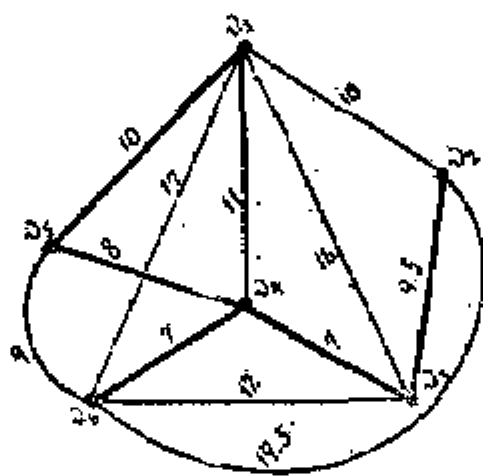


图 3—19

习 题

3—1　画出n个标号点的所有树；$n=1，2，3，4，5$。

3—2　画出n个无标号点的所有树；$n=1，2，3，4，5$。

3—3　证明一条路径是它自己的生成树。

3—4　证明当且仅当一个连通图G的任意一个子图g不含有回路时，g包含在G的某一棵生成树中。

3—5　证明汉密尔顿路径是一棵生成树。

3—6　证明图G的任何一个回路同一弦组至少有一条共同边。

3—7　如果给你一个图的所有生成树，你能构成一个图吗？怎样做？

3—8　不列出一个连通图的生成树，试寻找一个测定生成树总数的方法。

3—9　在一个给出的加权连通图G中，假定存在一条边e_s，它的权比G中其它边的权都要小。证明G中的每一棵最短生成树一定含有边e_s。

3—10　设G是一个加权连通图，在这个图中每一条边属于G中某一回路。如果e_l是G的所有边中数最大的边，证明G中没有一棵最短生成树包含e_l边的。

3—11　令T_1和T_2是连通图G的两棵生成树。如果边e属于T_1而不属于T_2，试证明存在另一条属于T_2而不属于T_1的边f，使得子图（$T_1 \ominus e$）$\cup f$和（$T_2 \ominus f$）$\cup e$都是G的生成树。

第四章 关联组和截割组

第一节 关 联 组

在我们研究线性图的时候，除了特殊说明外，我们假定图中是没有自环的。

在一个点上与它相关联或连接的所有边组叫作对应这个点的关联组，用符号$S(v)$来表示，括号中的字母v表示对应的点。根据这个定义，如果没有边关联到一个点v，那么对应这个点的关联组是空组。

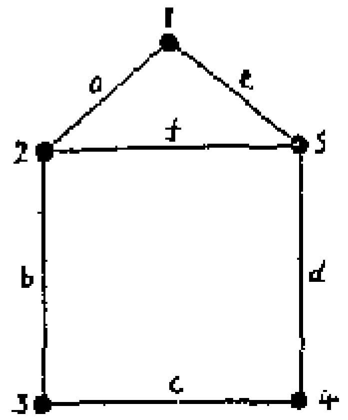


图 4－1

在一个图中，每一个点就有一个关联组。如以图4－1为例，它有5个点，因此这个图有5个关联组。对应着点1的关联组S（1）是（a，e），对应着点2的关联组S（2）是（a，b，f），其余对应着点3，4和5的关联组S（3），S（4）和S（5）分别为（b，c），（c，d）和（d，e，f）。

第二章中已经讲过，两个欧拉图的环和是一个欧拉图，而且两个回路的环和或是一个回路，或是一个边不共用回路的并集。可是两个关联组的环和，一般地讲，既不是一个关联组，也不是边不共用关联组的并集。为什么呢？假使我们考虑一个图G的两个关联组$S(v_1)$和$S(v_2)$，倘然$S(v_1)\oplus S(v_2)$是一个关联组的话，那么$S(v_1)\oplus S(v_2)$中的每一条边一定要关联到一个点v'的，而且在$S(v_1)\oplus S(v_2)$中的这些边一定是图G中只关联到点v'的那些边。换句话说，假使

$$S(v_1)=(e_1,e_2,\cdots,e_p,a_1,a_2,\cdots,a_m)$$

和

$$S(v_2)=(e_1,e_2,\cdots,e_p,b_1,b_2,\cdots,b_n)$$

要注意到，那些边$e_1,e_2,\cdots,e_p$是既在$S(v_1)$之中，又在$S(v_2)$之中的，那么

$$S(v_1)\oplus S(v_2)=(a_1,a_2,\cdots,a_m,b_1,b_2,\cdots,b_n)$$

既然边$a_1,a_2,\cdots,a_m$不在$S(v_2)$之中，它们是不与点v_2相关联的，那么$S(v_1)\oplus S(v_2)$也不是对应点v_2的关联组。同样，$S(v_1)\oplus S(v_2)$也不是对应点v_1的关联组。只有当$S(v_1)\oplus S(v_2)$中每一条边关联到除点v_1和v_2以外的一个点，并且原来给出的图中没有其它边关联到这一个点时，$S(v_1)\oplus S(v_2)$才是一个关联组。所以原来给出的图一定要有如图4－2（a）所示的那样的子图作为一个部分的。因此讲两个关联组的环和一般不是一个关联组。

那么，$S(v_1)\oplus S(v_2)$会不会是边不共用关联组的并集呢？如果$S(v_1)\oplus S(v_2)$没有共同边，那么很明显$S(v_1)\oplus S(v_2)$是一个边不共用关联组的并集。可是，这种情形是一个特例。所以我们在上面的讨论中假定$S(v_1)$和$S(v_2)$是有共同边的。要是$S(v_1)\oplus S(v_2)$是边不共用关联组的并集的话，$S(v_1)\oplus S(v_2)$一定可以分解成$S(v_1')$，$S(v_2')$，…，$S(v_k')$等k个组，其中$k\geqslant 2$。这样在$S(v_1)\oplus S(v_2)$中的每一条边正好是这些组中的一条边。而且，组$S(v_r')$一定是给出的图中对应于点v_r'的关联组，其中所有$r=1,2,\cdots,k$。因此原来给出的图一定包括如图4－2（b）所示那样的子图作为一个部分。这样，我们得出一个结论，一

般地讲，两个关联组的环和既不是一个关联组，也不是边不共用关联组的并集。

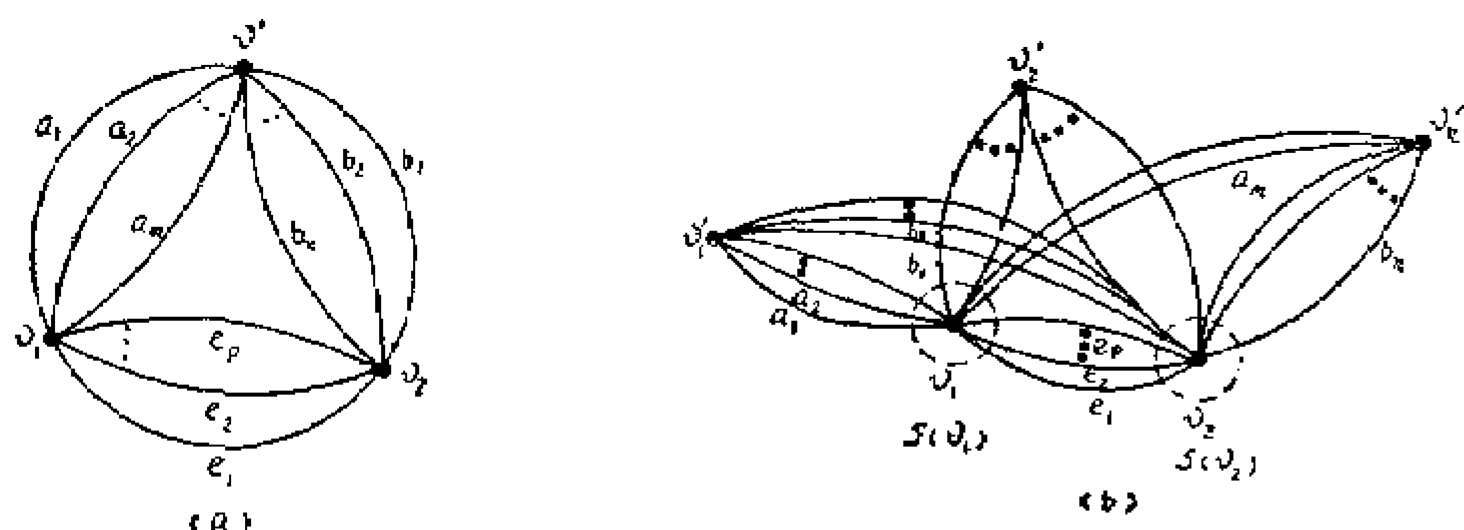


图 4—2

第二节　截割组的定义

在一个连通图G中，如果一个边组S从G中全部删除后，使图的秩数R正好减少1，而且在这个边组中没有一个真子组（所谓S的真子组是S中与S不恒等的子组）假如有这种性质的话，那么这个边组S就叫作截割组，或称割集。

例4—1　在图4—3（a）中的一个边组（b，c，d，f）中，删除这个边组中的所有边，使得图一分为二，成为一个分离图，如图（b）所示。图（a）的秩数R为$n_v-k=5-1=4$，而图（b）的R为$u_v-k=5-2=3$。如果在边组（b，c，d，f）中只删除其中几条边而不是所有边的话，譬如说只删除边b，c和d，使得原图成为一个如图（c）所示的一个连通图，它的秩数仍旧是4，即秩数并未减少。所以根据定义，边组（b，c，d，f）是一个截割组。由此可见，一个截割组总是把一个图一分为二的。

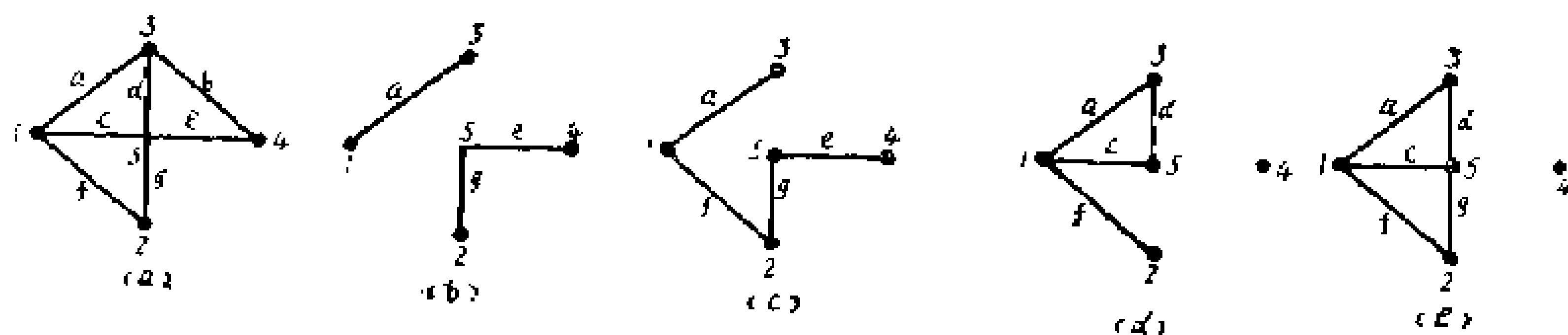


图 4—3

再看图4—3（a）中的一个边组（b，e，g）。把这个边组中所有的边删除后得图（d）。在第二章中已经讲过，边的删除并不意味着端点的删除，所以在图（d）中仍保留着点4，图（d）的秩数为$5-2=3$，即边组（b，e，g）的删除，确实使图（d）的秩数正好减一，而且这也是一个分离图。可是，在边组（b，e，g）中却有一个真子组（b，e），从图（a）中删除真子组（b，e）后得图（e），后者的秩数也比图（a）正好减少一，而且图（e）也是一个分离图。所以根据截割组的定义，边组（b，e，g）不能算作一个截割组，而边组（b，e）才是一个截割组。所以对一个截割组也可以这样下定义，即它是连通图中的一个最小边组，后者的删除使原图的秩数减少一。

例4－2　图4－4（a）是一个连通图，它的秩数为$6-1=5$。边组（a，c，d，f）的删除使图（a）成为图（b）那样的一个分离图，它的秩数为$6-2=4$，比图（a）正好减少1，而且边组（a，c，d，f）中没有其它子组有这样的性质，所以边组（a，c，d，f）是一个截割组。再看边l作为一个边组看，把它从图（a）中删除，使图（a）成为图（c），因为点5仍保留着，所以图（c）的秩数为4，比图（a）正好减少1。因此边l也是一个截割组。

在树中，任何一条边（即枝）的删除把一棵树一分为二，所以一棵树的每一条边是一个截割组。

在研究通信网络和运输网络的性质时，截割组是有重大意义的。譬如，图4－4（a）中的6个点代表6个城市，由电话线（即边）联系着，我们希望找出网络中的弱点，以便用外加的电话线来加强它。我们就要研究图中所有的截割组，凡是截割组中有最少边的那个组就是最脆弱的地方。在图4－4（a）中，点5所代表的那个城市，只要边l一切断，它就与其它城市失去联系。

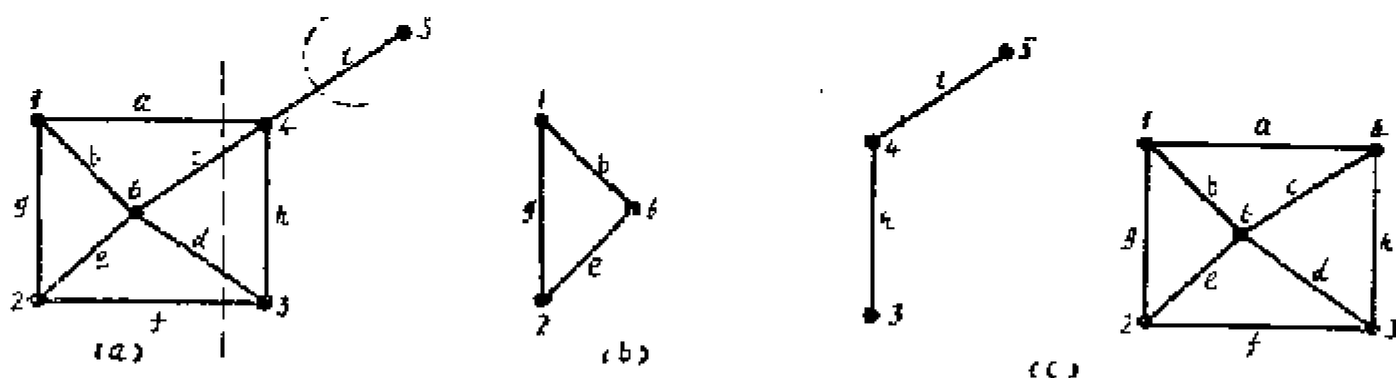


图　4－4

第三节　截割组的若干特性

试考虑一个连通图G中的一棵生成树T和G中任何一个截割组S。同一个图中，S与T是否会有任何共同边呢？回答是“有的”，否则从G中删除一个截割组S就不会把G一分为二了。

定理4－1：一个连通图G中每一个截割组必定至少含有G中每棵生成树的一个枝。

那么这个定理的逆定理是否也正确呢？换句话说，是否可以说一个连通图中一个最小边组中至少含有每棵生成树的一个枝的就是截割组呢？回答是“对”的，理由如下。

在一个给定的连通图G中，设Q是G中至少含有每一棵生成树的一枝的最小边组。（$G\ominus Q$）是G中删除Q后所余留的子图。既然子图（$G\ominus Q$）不含有G的生成树，所以（$G\ominus Q$）是分离的，其中有一部分可能只有一个孤立点。而且，由于Q是有这种性质的最小边组，如果把Q中任何一条边e还给（$G\ominus Q$），那么至少可以产生一棵生成树。这就是说（$G\ominus Q\oplus e$）是一个连通图。因此从G中删除一个最小边组Q可以把G一分为二。这就是截割组的定义。所以

定理4－2：在一个连通图G中，含有每一棵生成树中至少一枝的任何一个最小边组是一个截割组。

现在再提出一个重要定理。

定理4－3：每一个回路与任何一个截割组之间的共同边数为偶数。

证：考虑图 4 — 5 中图G的一个截割组S。

图 4 — 5

设从图G中删除S后，G中各点划分为两个互不相关的子组V_1和V_2。考虑在 G 中有一个回路C（如粗线所示）。如果C中所有各点完全在点组V_1（或V_2）之中，那么S和 C 之间的共同边数n_e就是零，即$n_e(S\cap C)=0$，0是一个偶数。另一方面，如果 C 有些点在 V_1 之中，有些点在V_2之中，当我们在点组V_1和V_2之间来回循着图 4 — 5 中粗线边中箭头所示方向沿着回路C穿行时，由于一个回路的闭合性质，在V_1和V_2之间所经过的边数一定是偶数的。还因为S中每一条边的一端在V_1中，而另一端在V_2中，而且 G 中的其它边没有这种性质，所以说S与C之间的共同边数为偶数。

第四节　基本截割组

在本章第二节中，已经讲到如何利用截割组来识别一个通信网络中的弱点。为此目的，我们要能够列举一个图的所有截割组，然后从中找出边数最少的那个截割组。即使如图4—4那样的简单图也有很多截割组。所以我们一定要有一个有规则的方法来产生所有有关的截割组。

在回路方面，也有相似的问题。因为一个图中的很多回路是两个或更多个基本回路的组合，所以为了找寻回路组，先要找出基本回路。在截割组方面，也可以用同样的办法。正因为一棵生成树对确定一个基本回路组是必要的，一棵生成树对确定一个基本截割组也是必要的。所以注意回路组与截割组之间的类似性对我们了解截割组是有好处的。

设一个连通图G中有一棵生成树T，b是T中的任何一枝。因为组（b）是T中的一个截割组，它把T中所有点划分为分离的两组，每组各在枝b的一端。再考虑G中各点的这个划分和对应这个划分的截割组S。截割组S将只含有T的一枝b，而 S 中其余（如果有的话）各边是T的弦。这种只含树T的一枝的一个截割组S叫作T的一个基本截割组。在图 4 — 6 中，有粗线所示的一棵生成树T以及虚线所示的T的所有 5 个基本截割组。

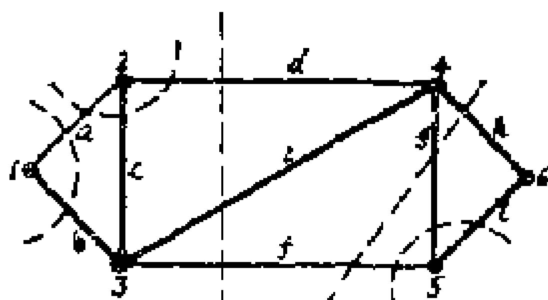


图 4 — 6

正象一棵生成树的每一条弦确定一个唯一的基本回路，一棵生成树的每一个枝确定一个唯一的基本截割组。必须注意，所谓基本截割组（正像基本回路一样）只是对某一给定的生成树才是有意义的。

设想一个连通图G中有一棵生成树T。设c_1是关于T的一条弦，并且设由c_1所组成的基本回路叫C，它除了有弦c_1外，还有k个枝b_1，b_2，…，b_k，就是说，$C=(c_1, b_1, b_2, \cdots, b_k)$是关于$T$的一个基本回路。任何一棵生成树的每一个枝与它相结合的有一个基本截割组。设S_1是与b_1相结合的基本截割组，它除了有枝b_1外，还有q条弦，就是说，$S_1=(b_1,$

c_1, c_2, …, c_q）是关于T的一个基本截割组。

根据定理4—3，在C与S_1之间有偶数条共同边。边b_1是既在c之中，又在S_1之中。那么只有另一条其它边在C中，这就是c_i，它也可能在S_1之中。所以这样就一定有两条边，即b_1和c_i是S_1和C所共有的，因此弦c_i是弦组c_1, c_2, …, c_q之中的一条。

这个论证对于与b_2, b_3, …, b_k相结合的基本截割组也是对的。所以弦c_i是含于C中与诸枝相结合的每一个基本截割组中。

那么c_i是否可能在于除那些与b_1, b_2, …, b_k相结合之外的任何别的基本截割组S'（当然对T而言的）之中呢？回答是否定的，否则既然C中设有一枝是在S'之中，那么只有一个结果，就是S'和C只有一条共同边c_i了。这是与定理4—3相违背的。因此我们得到一个重要结果，即

定理4—4：对一棵给定的生成树T而言，确定着一个基本回路C的一条弦c_i只出现在C中诸枝所确定的那些基本截割组之中，而不会出现在其它之中。

例4—3　图4—6中粗线所示是一棵生成树（b, c, e, k, l）。由弦f所组成的基本回路是（f, e, k, l）。由3个枝e, k和l所确定的3个基本截割组是

由枝e所确定的：（d, e, f）

由枝k所确定的：（f, g, k）

由枝l所确定的：（f, g, l）

弦f在这3个基本截割组中每个都有，但是其它基本截割组不会含有f的。

定理4—4的逆定理也是正确的。

定理4—5：对于一个给定的生成树T而言，确定着一个基本截割组S的一个枝b_i只出现在S中诸弦所确定的那些基本回路之中，而不会在其它之中。

证：这个证明所用的论证与定理4—4所用的相似。设由一个枝b_i所确定的基本截割组S为

$$S=(b_i, c_1, c_2, \cdots, c_p)$$

并设由弦c_1所确定的基本回路C_1为

$$C_1=(c_1, b_1, b_2, \cdots, b_q)$$

因为S和C_1的共同边数一定是偶数，b_i一定是在C_1之中。对于由弦c_2, c_3, …, c_p所组成的基本回路，这也是正确的。

另一方面，假使b_i出现在由弦c_1, c_2, …, c_p以外的一条弦所组成的一个基本回路C_{p+1}之中。既然诸弦c_1, c_2, …, c_p之中没有一条弦是在C_{p+1}之中，那么对于一个回路C_{p+1}与截割组S之间只有一条共同边b_i了，这是不可能的。这就证明了这个定理。▌

例4—4　以图4—6中生成树（b, c, e, k, l）中的枝e为例，由e所确定的基本截割组是（d, e, f），由弦d和f所确定的两个基本回路是

由弦d所确定的：（c, d, e）

由弦f所确定的：（e, f, k, l）

枝e都包括在这两个基本回路中，而在其它三个基本回路中都不含有枝e的。

第五节　截割组的环和

在研究截割组的环和之前，让我们先讨论一下边组的分解。

设一个图中点组V的非空不共用真子组为V_a，V_b，V_c和V_d。就是

$$V_p \subset V \quad (4-1)$$

$$V_p \neq \phi \quad (4-2)$$

式中$p \in (a, b, c, d)$，并且当p，$q \in (a, b, c, d)$和$p \neq q$时

$$V_p \cap V_q = \phi \quad (4-3)$$

设：

$$V_A = V_a \cup V_b \quad (4-4)$$

和

$$V_B = V_c \cup V_d \quad (4-5)$$

则

$$E(V_A \times V_B) = E(V_a \cup V_b \times V_c \cup V_d) \quad (4-6)$$

其中E为边组，符号$E(V_A \times V_B)$表示一个边组中每一条边从V_A中的一个点连接到V_B中的一个点。我们可以把$E(V_a \cup V_b \times V_c \cup V_d)$分解成下列4组。

（一）$E(V_a \times V_c)$，包括V_a中的一个点与V_c中的一个点之间相连接的所有那些边。

（二）$E(V_a \times V_d)$，包括V_a中的一个点与V_d中的一个点之间相连接的所有那些边。

（三）$E(V_b \times V_c)$，包括V_b中的一个点与V_c中的一个点之间相连接的所有那些边。

（四）$E(V_b \times V_d)$，包括V_b中的一个点与V_d中的一个点之间相连接的所有那些边。

换句话说，$E(V_a \cup V_b \times V_c \cup V_d)$可以表示为

$$E(V_a \cup V_b \times V_c \cup V_d) = E(V_a \times V_c) \cup E(V_a \times V_d) \cup E(V_b \times V_c) \cup E(V_b \times V_d) \quad (4-7)$$

由此可见，这些边组$E(V_a \times V_c)$，$E(V_a \times V_d)$，$E(V_b \times V_c)$和$E(V_b \times V_d)$中任何两组间都没有共同边。所以式（4—7）可以写成为：

$$E(V_a \cup V_b \times V_c \cup V_d) = E(V_a \times V_c) \oplus E(V_a \times V_d) \oplus E(V_b \times V_c) \oplus E(V_b \times V_d) \quad (4-8)$$

同样，$E(V_A \times V_A)$可以展开为：

$$E(V_A \times V_A) = E(V_a \cup V_b \times V_a \cup V_b)$$
$$= E(V_a \times V_a) \oplus E(V_a \times V_b) \oplus E(V_b \times V_b) \quad (4-9)$$

但是由于这里所考虑的线性图是无向的，所以

$$E(V_a \times V_b) = E(V_b \times V_a) \quad (4-10)$$

现在我们来研究如何从一个图的已知截割组来求得其它截割组。先研究一个定理。

定理4—6：一个图中任何两个截割组的环和是另一个截割组，或者是些截割组的边不共用并集。

证：设S_1和S_2是一个连通图G的两个截割组，V_1和V_2是G的点组V中对应S_1的两个非空不共用真子组，V_3和V_4是对应S_2的两个非空不共用真子组。两个截割组和它们的点组V划分情形如图4—7所示。由图（a）和图（b）可见

$$V_1 \cup V_2 = V \text{和} V_1 \cap V_2 = \phi$$

$$V_3 \cup V_4 = V \text{和} V_3 \cap V_4 = \phi$$

设子组$(V_1 \cap V_4) \cup (V_2 \cap V_3)$叫作$V_5$。根据定义，这个子组等于$V_1 \oplus V_3$的。同样，设子组$(V_1 \cap V_3) \cup (V_2 \cap V_4)$叫作$V_6$，这等于$V_2 \oplus V_3$的。它们的关系见图4—7（$c$）。

两个截割组的环和S_1和S_2只包括连接V_5中的点到V_6中的点的那些边。而且在$S_1 \oplus S_2$之外是没有连接V_5中的点到V_6中的点的边的。

因此$S_1 \oplus S_2$这个边组把V划分为V_5和V_6，即$V_5 \cup V_6 = V$和$V_5 \cap V_6 = \phi$。

所以从G中删除$S_1 \oplus S_2$后，如果每个包括V_5和V_6的子图仍旧是连通的话，那么$S_1 \oplus S_2$

是一个截割组，否则就是些截割组的一个边不共用并集。

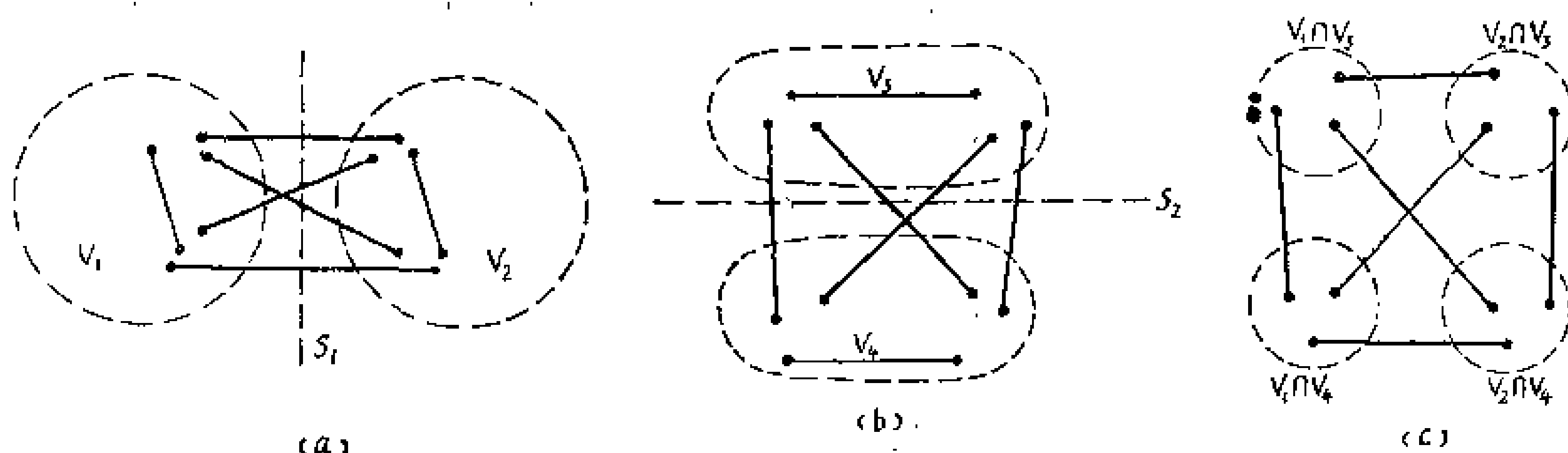


图 4—7

例4—5　在图4—6中，让我们求下列3对截割组的环和

$(d,e,f)\oplus(f,g,k)=(d,e,g,k)$另一个截割组

$(a,b)\oplus(b,c,e,f)=(a,c,e,f)$另一个截割组

$(d,e,g,k)\oplus(f,g,l)=(d,e,f,k,l)$

$=(d,e,f)\cup(k,l)$

两个截割组的一个边不共用并集

由此可见，由一个图中若干已知截割组可以得到一个产生其它截割组的方法。不过用这个方法，我们是不能从图中任何两个截割组开始而得到图中所有截割组的。我们必须从前一节中所讲的关于一个已知生成树的所有基本截割组着手，才能得到图中所有的截割组。

例4—6　在图4—8中，粗线所示是这个图的一棵生成树，关于这棵生成树的基本截割组有：$S_1=(a,c,f)$，$S_2=(a,d,e)$，$S_3=(b,e)$和$S_4=(a,c,g)$。通过环和运算，我们可以得到这个图的所有截割组，截割组的边不共用并集，和空组的集体$\{S\}$的所有元如下：

$S_1\oplus S_2=(c,d,e,f)$

$S_1\oplus S_3=(a,b,c,e,f)$

$S_1\oplus S_4=(f,g)$

$S_2\oplus S_3=(a,b,d)$

$S_2\oplus S_4=(c,d,e,g)$

$S_3\oplus S_4=(a,b,c,e,g)$

$S_1\oplus S_2\oplus S_3=(b,c,d,f)$

$S_1\oplus S_2\oplus S_4=(a,d,e,f,g)$

$S_1\oplus S_3\oplus S_4=(b,e,f,g)$

$S_2\oplus S_3\oplus S_4=(b,c,d,g)$

$S_1\oplus S_2\oplus S_3\oplus S_4=(a,b,d,f,g)$

$S_1\oplus S_1=S_2\oplus S_2=S_3\oplus S_3=S_4\oplus S_4=\phi$

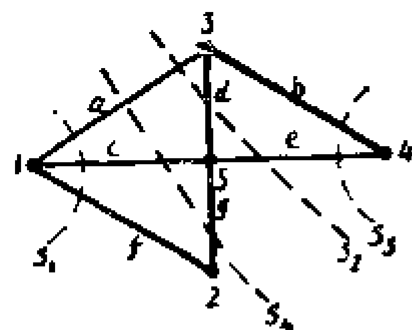


图 4—8

注意上述运算结果中，$S_1\oplus S_3, S_3\oplus S_4, S_1\oplus S_2\oplus S_4$和$S_1\oplus S_2\oplus S_3\oplus S_4$都是些截割组的边不共用并集。

关于截割组与关联组的关系有下列几个重要定理。

定理4—7：任何一个截割组可以表示为关联组的环和。

定理4—8：设G为一个连通图，则任何一个关联组等于G中所有其余关联组的环和。

这两个定理的证明从略。

定理4—9：设G是一个包括n_v个点和k个部分的分离线性图。则它正好有n_v-k个线性无关关联组。

根据定理4—8，一个线性图的n_v-k个线性无关关联组通过环和运算就足以求得这个图的所有截割组。换句话说，一个图的所有截割组，截割组的边不共用并集，和空集的集体$\{S\}$可以表达为

$$\{S\}=\{S(v_{i1})\oplus S(v_{i2})\oplus\cdots\oplus S(v_{ir})\}$$

$$S(v_{i1}),S(v_{i2}),\cdots,S(v_{ir})\in\{S(v)\},\quad r=1,2,\cdots,n_v-k \qquad (4—11)$$

式中$\{S(v)\}$是图的n_v-k个线性无关关联组的集体。

例4—7　在图4—8中，因为$k=1$，即$n_v-k=4$，所以这个图有4个线性无关关联组。假使我们选$S(1)=(a,c,f)$，$S(3)=(a,b,d)$，$S(4)=(b,e)$和$S(5)=(c,d,e,g)$作为4个线性无关关联组，那么

$S(1)\oplus S(3)=(b,c,d,f)$

$S(1)\oplus S(4)=(a,b,c,e,f)$

$S(1)\oplus S(5)=(a,d,e,f,g)$

$S(3)\oplus S(4)=(a,d,e)$

$S(3)\oplus S(5)=(a,b,c,e,g)$

$S(4)\oplus S(5)=(b,c,d,g)$

$S(1)\oplus S(3)\oplus S(4)=(c,d,e,f)$

$S(1)\oplus S(3)\oplus S(5)=(b,e,f,g)$

$S(1)\oplus S(4)\oplus S(5)=(a,b,d,f,g)$

$S(3)\oplus S(4)\oplus S(5)=(a,c,g)$

$S(1)\oplus S(3)\oplus S(4)\oplus S(5)=(f,g)$

再加$S(1)$，$S(3)$，$S(4)$，$S(5)$和空组就是$\{S\}$中的所有组。

这里所得结果是与例4—6所得的结果完全相同的。

第六节　连通度和可分性

一个连通图G的每一个截割组包括若干条边。最小截割组（即有最少边数的截割组）的边数叫作G的边连通度，也可以把一个图的边连通度解释为从图中删除的最少边数而能使图的秩数正好减少1。例如，一棵树的边连通度为1。图4—4（a），4—6和4—8的边连通度分别为1、2和2。

观察图4—9，我们可以发现从这个图中删除一条边，甚至两条边，也不能使这个图分离为二。可是删除一个点v却能把这个图分割为二个图。一个连通图的点连通度，简称连通度，就是从图中删除最少点数而能使连接图分离成为二个图的。一棵树的点连通度为1。图4—4（a）、4—6和4—9的点连通度分别为1、2和1。我们这里对边连通度和点连通度下定义时，都是对连通图而言的。有的书上把分离图的边连通度和点连通度都说是零。

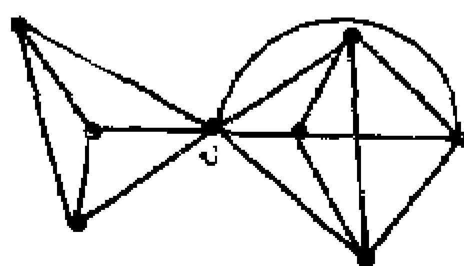


图　4—9

如果一个连通图的点连通度为 1 的这个图就称为可分图。这个问题已经在第二章中讲过。可分图也可以这样下定义，就是，如果一个连通图G有一个子图 g，它与余图$\bar{g}$之间只有一个共同点，这个连通图G就叫作可分图。图 4 — 9 中的点v叫作截割点。在一棵树中，凡是点的次数大于一的都是截割点。关于截割点有下面的定理。

定理 4 —10：在一个连通图G中，当且仅当G中有这样的两点x和y，它们之间的每一条路径必须通过一个点v，这个点v就是G的截割点。

这个定理的含义是很重要的。对通信网络而言，如果这个通信网络用一个线性图G来表示，这个点v就是x和y之间必经的关键点。所以连通度和可分性的研究是有实际应用意义的。现在举例来说明它们的应用。

假设我们有n_v个站用n_e条线路来连通。这些线路可能是电话线、桥梁、铁路、公路、隧道、或输油管等。其中$n_e \geqslant n_v - 1$。这些站之间用怎样的连接法最好呢？所谓“最好”是指这个网络中某一个站或某一条线路，或某些站或某些线路遭受破坏时，网络要尽量少受损害。换句话说，由n_v个点和n_e条边所构成的图要有最大可能的边连通度和点连通度。

例如，图 4 — 9 中有$n_v = 8$，和$n_e = 16$，这个图的点连通度为 1 和边连通度为 3。用相同的点数和边数（即$n_v = 8$，$n_e = 16$）构成的另一个图如图 4 —10所示，而这个图的点连通度和边连通度都是 4。因此，这个系统中即使有 3 个站被炸或 3 条线路被破坏，其余各站相互之间仍旧能够联系。所以如图 4 —10的网络虽然仍旧用16条线路来连通 8 个站，要比图 4 — 9 那样的系统好得多。

在给定的n_v和n_e的条件下，什么是最高的点连通度和边连度呢？下面那些定理是答案。

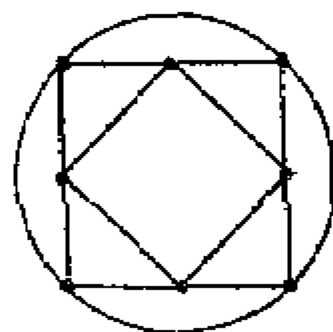


图 4 —10

定理 4 —11：一个图G的边连通度不可能超过G中最小关联组的次数。

证：设点v_i为图G中的最少次数的点。设$d(v_i)$是点v_i的次数。从图G中删除关联到点v_i的$d(v_i)$条边，就能使点v_i从G中分离出来。所以最小关联组的次数就是图G的边连通度。▌

定理 4 —12：任何图G的点连通度永远不能超过G的边连通度。

证：设G的边连通度为α。G中就有一个截割组S是关联着α条边的。设S划分G的所有点为V_1和V_2两个子组。靠从S中的那些边所关联的V_1（或V_2）中至多删除α个点时，我们就能达到从G中删除S，连同这些点中关联着的所有其它边。这就证明了这个定理。▌

由此得到一个推论：在一个有两个点以上的不可分图中，每个截割组至少含有两条边。

定理 4 —13：在一个有n_v个点和n_e条边（$n_e \geqslant n_v - 1$）的图G中所能得到的最大点连通度为$2n_e/n_v$的整数部分，即$\lfloor 2n_e/n_v \rfloor$。

这是因为G中每一条边提供的次数为 2。一个图的总次数为$2n_e$，分配给n_v个点。既然$n_e \geqslant n_v - 1$，所以图G中一定至少有一个点的次数等于或小于$2n_e/n_v$。根据定理 4 —11 和定理 4 —12，G的点连通度不能超过这个数$2n_e/n_v$。

为了继续进行证明，一个人可以先画出一个有n_v个点和次数为$\lfloor 2n_e/n_v \rfloor$的正则图，然后任意添加其余的$n_e - (n_v/2) \cdot \lfloor 2n_e/n_v \rfloor$条边。读者试完成其余的证明。

上述 3 个定理可以概括如下：

$$点连通度 \leqslant 边连通度 \leqslant 2n_e/n_v \qquad (4—12)$$

$$最大可能的点连通度 = \lfloor 2n_e/n_v \rfloor \qquad (4—13)$$

所以，如图4－9和4－10所示的有8个点和16条边的图的最大可能的点连通度为$2\times16/8=4$。

如果一个图G的点连通度为k，这个图就叫作k-连接图。所以，一个所谓1-连接图就是说它是一个可分图。

除了起点和终点之外，路径之间没有共同点的叫作不相交路径或点不共用路径。

定理4－14：当且仅当一个连通图G的每一对点之间是由k或更多的不相交路径所连接，并且至少有一对点是正好由k个不相交路径连接时，G是一个k-连接图。

定理4－15：当且仅当一个图G的每一对点是由k或更多的边不共用路径（路径可能相交，但是没有共同边）所连接，并且至少有一对点是正好由k个边不共用路径所连接时，这个图G的边连通度为k。

习　题

4－1　挑选图4－10中任意一棵生成树，列出关于这棵树的7个基本截割组。

4－2　取题4－1中7个基本截割组，利用环和运算得出这个图的所有截割组。

4－3　列出图4－4（a）中关于点对2，3之间的所有截割组。

4－4　证明图2－12中图的边连通度和点连通度都等于3。

4－5　n个点完全图的边连通度是多少？

4－6　证明在一个连通图G中，G中的一个截割组的余图不含有生成树，一棵生成树的余图（即弦组）不含有截割组。

4－7　证明在一个不可分图G中，关联到G的每个点的边组是一个截割组。

4－8　证明具有3个或更多点的一个连通图中，至少存在2个点不是截割点。

4－9　证明当且仅当图G的每个点都能够放在G的某一回路时，图G是不可分图。

4－10　在一个连通图G中，设Q是具有以下特性的一个边组：

（1）Q同G的每一截割组有偶数的共同边（包括零）；

（2）不存在满足特性（1）的Q的真子集。

证明Q是一个回路。

4－11　画出一个具有下列特性的图G：G的边连通度是4，点连通度是3，每点的次数$\geqslant5$。

4－12　证明一个欧拉图不可能具有奇数边的截割组。

第五章　图的矩阵表示法

第一节　矩阵的基本运算

在讲到图的矩阵表示法之前，我们先复习一下矩阵的基本概念和几种基本运算。

由$m\times n$个实数或复数排成m行和n列的表叫作$m\times n$矩阵A。例如

$$A=\begin{bmatrix} a_{11} & a_{12}\cdots & a_{1n} \\ a_{21} & a_{22}\cdots & a_{2n} \\ \vdots & \vdots & \vdots \\ a_{m1} & a_{m2}\cdots & a_{mn} \end{bmatrix} \qquad (5-1)$$

当$m=n$时，A就叫方矩阵，或叫n阶方阵，或简称方阵。

在矩阵中，a_{ij}是行i和列j的元素。作为一种简单表示法，式（5－1）可以写成

$$A=[a_{ij}];\quad i=1,2,\cdots,m$$
$$j=1,2,\cdots,n \qquad (5-2)$$

一个由$m\times 1$个数排成的矩阵叫作列矩阵。

例如

$$X=\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \qquad (5-3)$$

同样，一个由$1\times n$个数排成的矩阵叫作行矩阵。例如

$$X=[x_1\quad x_2\ \cdots\ x_n] \qquad (5-4)$$

一个方阵$A=[a_{ij}]$具有下列性质，即

$$a_{ij}=0,\quad i\neq j \qquad (5-5)$$

叫做对角矩阵。就是说，除了主对角线的元素外，其它的元素都是零，如

$$A=\begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix} \text{或}\quad A=\begin{bmatrix} a_{11} & & 0 \\ & a_{22} & \\ 0 & & \ddots \\ & & & a_{nn} \end{bmatrix} \qquad (5-6)$$

对角矩阵有时用符号$diag$来表示。

一个单位矩阵U是一种对角矩阵，它的$a_{ii}=1$；$i=1,2,\cdots,n$，如

$$U=\begin{bmatrix}1 & 0 & \cdots & 0\\ 0 & 1 & \cdots & 0\\ \vdots & \vdots & & \vdots\\ 0 & 0 & \cdots & 1\end{bmatrix} \text{或} \quad U=\begin{bmatrix}1 & & & \\ & 1 & & \mathbf{0}\\ & \mathbf{0} & \ddots & \\ & & & 1\end{bmatrix} \qquad (5-7)$$

就是说，主对角线上的元素都是1，而其余元素都是0。

如果一个矩阵A的$a_{ij}=a_{ji}$，这个矩阵叫作对称矩阵。

一个置换矩阵是一种矩阵，在每行（或列）中只有一个元素的值不是0而是1。在N阶的置换矩阵中，有$N!$个置换矩阵。

如果一个矩阵的所有元素都是零，这个矩阵叫作零矩阵，用[0]来表示。

当且仅当两个矩阵$A=[a_{ij}]$和$B=[b_{ij}]$的对应元素都是相等时，即

$$a_{ij}=b_{ij}; \quad i=1,\ 2,\ \cdots,\ m, \quad j=1,\ 2,\ \cdots,\ n,$$

A和B是相等的。这就在定义中隐含有A和B必须有相同的行数和列数，即是同阶的。也就是如果A是$m\times n$，则B一定也是$m\times n$。

如果两个矩阵$A=[a_{ij}]$和$B=[b_{ij}]$是同阶的，它们的和与差是

$$A\pm B=C=[C_{ij}] \qquad (5-8)$$

其中：$c_{ij}=a_{ij}\pm b_{ij}$

矩阵的加法是符合交换律的，即

$$A+B=B+A \qquad (5-9)$$

同样，加法也是符合结合律的，即

$$A+(B+C)=(A+B)+C \qquad (5-10)$$

当一个矩阵A乘以一个常数或纯量k时，等于A中每一元素a_{ij}乘以K，即

$$KA=[Ka_{ij}] \qquad (5-11)$$

设矩阵A为$m\times n$，矩阵B为$n\times p$，则当A左乘以B时所得的乘积AB为

$$AB=C=[c_{ij}] \qquad (5-12)$$

其中：$c_{ij}=\sum_{k=1}^{n}a_{ik}b_{kj}$; $i=1,\ 2,\ \cdots,\ m$, $j=1,\ 2,\ \cdots,p$，结果矩阵C是$m\times p$。

乘法的运算是行乘列运算。就是说，A中行i的每一元素乘以B中列j的对应元素。这些乘积合起来得到C中的c_{ij}。所以两个矩阵A和B相乘时，必须是A的列数等于B的行数。应注意矩阵的乘法一般不适合交换律，即

$$AB\neq BA \qquad (5-13)$$

所以矩阵A乘以矩阵B，必须说明是左乘还是右乘。

矩阵的乘法适合下列诸定律：

（一）结合律　$A(BC)=(AB)C$　　(5—14)

（二）分配律　$A(B+C)=AB+AC$　　(5—15)

$(A+B)C=AC+BC$　　(5—16)

当一个矩阵A乘以一个同阶的单位矩阵U时，

$$AU=A \qquad (5-17)$$

如果A和B是两个方矩阵，而且

$$AB=BA=U \qquad (5-18)$$

那么B是A的逆矩阵，用A^{-1}来表示。任意一个方矩阵不一定有逆。矩阵有逆的条件是要求矩阵的行列式值$|A|$不等于零，即矩阵A是非奇异的。

由式（5—18）可知，A和B是互逆的。所以式（5—18）可以写成

$$AA^{-1}=A^{-1}A=U \tag{5—19}$$

逆矩阵在矩阵代数中起着普通代数中除法的作用。所以，如果

$$AB=CD \tag{5—20}$$

式中 A是一个非奇异矩阵，那么

$$A^{-1}AB=A^{-1}CD$$

或

$$B=A^{-1}CD \tag{5—21}$$

这就是矩阵的除法。

设一个$m\times n$矩阵A为

$$A=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \tag{5—22}$$

把它的行和列互换，如

$$A'=\begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix} \tag{5—23}$$

A'叫作A的转置矩阵。

矩阵的转置运算适合下列算律：

$$(A')'-A \tag{5—24}$$

$$(kA)'=KA' \tag{5—25}$$

$$(A+B)'=A'+B' \tag{5—26}$$

$$(AB)'=B'A' \tag{5—27}$$

$$(A_1A_2\cdots A_n)'=A_n'A_{n-1}'\cdots A_2'A_1' \tag{5—28}$$

矩阵A的初等变换有：

（一）对换矩阵中的两行或两列；

（二）用一个不是零的数乘矩阵的一行或一列；

（三）用一个数乘A的一行（或列）加到A的另一行（或列）。

矩阵经过初等变换后不改变它的秩。

一个矩阵可以用一些从左到右的横线和一些从上到下的竖线分成若干个子矩阵，这种方法叫做矩阵的分块。例如，矩阵A可以分块如下：

$$A=\left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] \tag{5—29}$$

同样，我们也可以只用横线把一个矩阵B分块成

$$B=\begin{bmatrix} B_{11} \\ \cdots\cdots \\ B_{21} \end{bmatrix} \tag{5—30}$$

则

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}\begin{bmatrix} B_{11} \\ B_{21} \end{bmatrix}=\begin{bmatrix} A_{11}B_{11}+A_{12}B_{21} \\ A_{21}B_{11}+A_{22}B_{21} \end{bmatrix} \tag{5—31}$$

在图论中所用的矩阵一般只含有两个元素值，即0和1。这种矩阵称为二进制矩阵，或（0，1）矩阵。它的运算有两种方式，就是模2算术和布尔算术。

在模2算术的运算中是把正常算术的结果除以2，弃其商数而只取其余数作为结果。根据这个定义，模2算术的规定为：

$$\begin{aligned} 1\times1&=1 \qquad & 1+1&=0 \\ 1\times0&=0 \qquad & 1+0&=1 \\ 0\times0&=0 \qquad & 0+0&=0 \end{aligned}$$

从这些规定可以看出，它与正常算术的区别只在于$1+1$，它的结果是0而不是2。模2算术中另一特殊规则是

$$-1=1$$

在布尔算术中，算符$+$可以看作是并集，而算符$\times$代表交集。布尔算术的规则为

$$\begin{aligned} 1\times1&=1 \qquad & 1+1&=1 \\ 1\times0&=0 \qquad & 1+0&=1 \\ 0\times0&=0 \qquad & 0+0&=0 \end{aligned}$$

由此可见，在这两种算术中唯一的区别只在于$1+1$的结果。

例5—1 求矩阵A与矩阵B之积C，其中

$$A=\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \qquad B=\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$$

如用模2算术

$$C=AB=\begin{bmatrix} 1\times1+0\times0+1\times1 & 1\times0+0\times1+1\times1 \\ 1\times1+1\times0+0\times1 & 1\times0+1\times1+0\times1 \end{bmatrix}$$

$$=\begin{bmatrix} 1+0+1 & 0+0+1 \\ 1+0+0 & 0+1+0 \end{bmatrix}$$

$$=\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

如用布尔算术，则结果为

$$C=\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}$$

第二节 关联矩阵

除了用组或集的概念来代表一个线性图外，我们也可以用矩阵来代表它，尤其是为了计算机处理，矩阵是一种方便和有用的手段。此外还可以利用矩阵代数和某些规则来研究图的结构特性。

设一个图G有n_v个点和n_e条边，但没有自环。我们可以作出一个$n_v\times n_e$矩阵，它的n_v行对应着n_v个点，它的n_e列对应着n_e条边。矩阵A的定义如下：

$$A=[a_{ij}] \tag{5—32}$$

式中 $a_{ij}=1$，如果边e_j关联到点v_i；

$a_{ij}=0$，e_j和v_i互不关联。

这样一个矩阵叫作点-边关联矩阵，或简称关联矩阵。一个图的矩阵A有时也可以写作$A(G)$。图5—1中有一个图，它的关联矩阵为

$$A(G)=\begin{array}{c|cccccccc|} & a & b & c & d & e & f & g & h\\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0\\ 2 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1\\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\\ 4 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0\\ 5 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0\\ 6 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0\end{array} \tag{5—33}$$

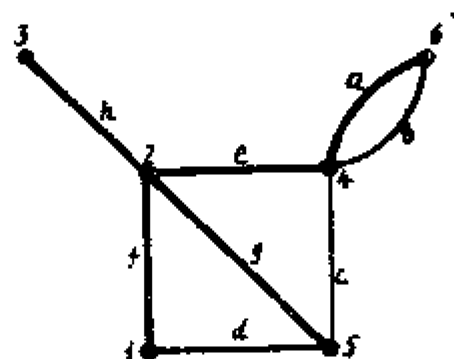


图 5—1

从一个没有自环的图很容易写出它的关联矩阵$A(G)$；另一方面，如果我们有了一个关联矩阵$A(G)$，我们也可以唯一地作出一个几何图形G来。关联矩阵和几何图形含有相同的信息，它们只不过是代表同一抽象图的不同方式而已。

从图5—1的矩阵中可以看到，关联组$S(1)$是由行1中的（d，f）表明的。关联组$S(2)$是由行2的（e，f，g，h）表明的，余类推。这个矩阵$A(G)$的每一行表示一个点以及对应这个点的关联组。

观察关联矩阵$A(G)$，我们得到下列信息：

（一）既然每条边只关联到两个点，A的每列正好只有两个1。

（二）每一行的1的数目等于对应点的次数。

（三）都是0的一行代表一个孤立点。

（四）一个图中的平行边在关联矩阵中表现为同样的列，例如，矩阵$A(G)$中的列1和2。

（五）如果一个图G是分离的，并且包括两个部分g_1和g_2，则图G的关联矩阵$A(G)$可以写成分块对角矩阵形式，如

$$A(G)=\begin{bmatrix}A(g_1) & 0\\0 & A(g_2)\end{bmatrix} \tag{5—34}$$

其中$A(g_1)$和$A(g_2)$分别是部分g_1和g_2的关联矩阵。这是因为g_1中的边不会关联到g_2中的点，反过来也是这样。这个说法对于一个图有任何部分也是正确的。

（六）一个关联矩阵中两行或两列的置换或排列相当于同一图中点和边的重新编号。从这一条就引出下一定理。

定理 5 — 1：当且仅当两个图G_1和G_2的关联矩阵$A(G_1)$和$A(G_2)$只区别于行和列的置换时，这两个图是同构的。

例 5 — 1　图 5 — 2 中有两个图G_1和G_2。图（a）中G_1的关联矩阵为

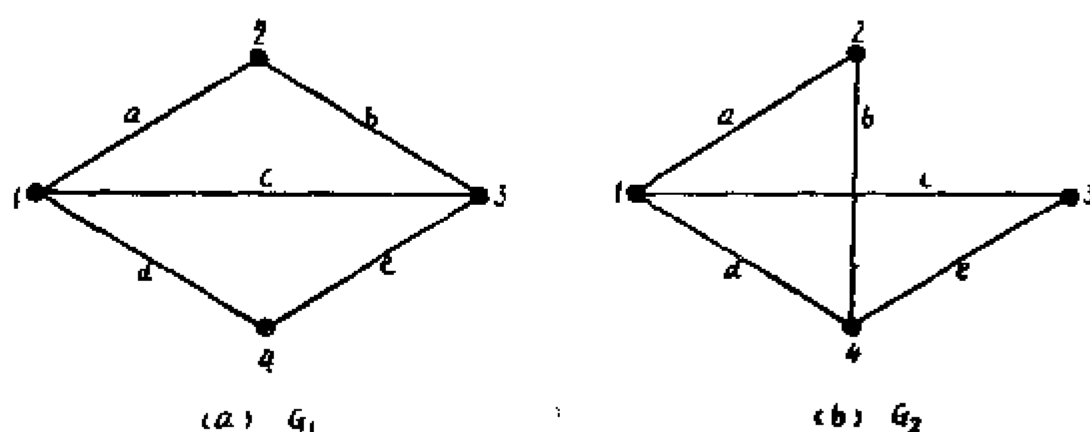


图 5 — 2

$$A(G_1)=\begin{array}{c}\\1\\2\\3\\4\end{array}\begin{array}{c}\begin{matrix}a&b&c&d&e\end{matrix}\\\begin{bmatrix}1&0&1&1&0\\1&1&0&0&0\\0&1&1&0&1\\0&0&0&1&1\end{bmatrix}\end{array}$$

把行 3 和行 4 互换得

$$\begin{array}{c}\\1\\2\\3\\4\end{array}\begin{array}{c}\begin{matrix}a&b&c&d&e\end{matrix}\\\begin{bmatrix}1&0&1&1&0\\1&1&0&0&0\\0&0&0&1&1\\0&1&1&0&1\end{bmatrix}\end{array}$$

现在再把列 c 和 d 互换得

$$A(G_2)=\begin{array}{c}\\1\\2\\3\\4\end{array}\begin{array}{c}\begin{matrix}a&b&c&d&e\end{matrix}\\\begin{bmatrix}1&0&1&1&0\\1&1&0&0&0\\0&0&1&0&1\\0&1&0&1&1\end{bmatrix}\end{array}$$

这就是图（b）G_2的关联矩阵。所以G_1和G_2是同构的。

关于一个图 G 的关联矩阵$A(G)$的秩数有下列定理。

定理 5 — 2：一个关联矩阵的秩数等于它的线性图的秩数。

所以一个连通图的关联矩阵的秩数是n_v-1，其中n_v是这个图的点数。一个分离图的关联矩阵的秩数是n_v-k，其中n_v是这个图的点数，k是部分数。

例 5 — 2　如图 5 — 3 所示的线性图，它的关联矩阵为

$$A(G)=\begin{array}{c}\\1\\2\\3\\4\\5\end{array}\begin{array}{c}\begin{array}{ccccccc}a&b&c&d&e&f&g\end{array}\\\left[\begin{array}{ccccccc}0&0&0&0&0&1&1\\0&0&0&1&1&1&0\\0&1&1&1&0&0&0\\1&1&0&0&0&0&0\\1&0&1&0&1&0&1\end{array}\right]\end{array}\qquad(5-35)$$

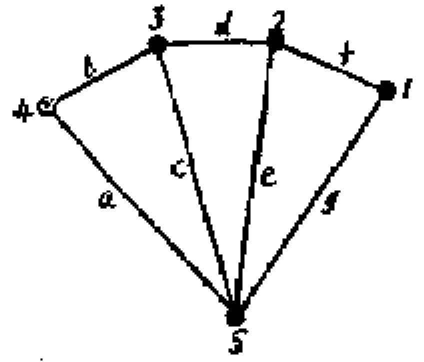


图 5 — 3

置换列 a 和列 g，并将行 1 加到行 5。因为二进制矩阵用模 2 算法，得

$$\begin{array}{c|c:cccccc}
 & g & b & c & d & e & f & a\\
\hline
1 & 1 & 0 & 0 & 0 & 0 & 1 & 0\\
\hdashline
2 & 0 & 0 & 0 & 1 & 1 & 1 & 0\\
3 & 0 & 1 & 1 & 1 & 0 & 0 & 0\\
4 & 0 & 1 & 0 & 0 & 0 & 0 & 1\\
5 & 0 & 0 & 1 & 0 & 1 & 1 & 1
\end{array}$$

置换列 b 和 d，并且将行 2 加到行 3，得

$$\begin{array}{c|cc:ccccc}
 & g & d & c & b & e & f & a\\
\hline
1 & 1 & 0 & 0 & 0 & 0 & 1 & 0\\
2 & 0 & 1 & 0 & 0 & 1 & 1 & 0\\
\hdashline
3 & 0 & 0 & 1 & 1 & 1 & 1 & 0\\
4 & 0 & 0 & 0 & 1 & 0 & 0 & 1\\
5 & 0 & 0 & 1 & 0 & 1 & 1 & 1
\end{array}$$

将行 3 加到行 5，得

$$\begin{array}{c|ccc:cccc}
 & g & d & c & b & e & f & a\\
\hline
1 & 1 & 0 & 0 & 0 & 0 & 1 & 0\\
2 & 0 & 1 & 0 & 0 & 1 & 1 & 0\\
3 & 0 & 0 & 1 & 1 & 1 & 1 & 0\\
\hdashline
4 & 0 & 0 & 0 & 1 & 0 & 0 & 1\\
5 & 0 & 0 & 0 & 1 & 0 & 0 & 1
\end{array}$$

最后将行 4 加到行 5，得

$$\begin{array}{c|cccc:ccc}
 & g & d & c & b & e & f & a\\
\hline
1 & 1 & 0 & 0 & 0 & 0 & 1 & 0\\
2 & 0 & 1 & 0 & 0 & 1 & 1 & 0\\
3 & 0 & 0 & 1 & 1 & 1 & 1 & 0\\
4 & 0 & 0 & 0 & 1 & 0 & 0 & 1\\
\hdashline
5 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

由这个矩阵知道关联矩阵的秩数为4，而图（a）的秩数为$n_v-1=5-1=4$。

如果我们从一个连通图的关联矩阵中移去任何一行，所余留下来的$(n_v-1)\times n_e$子矩阵的秩数仍是n_v-1。因为这(n_v-1)行中包含着整个矩阵的信息量，所以我们只需要一个关联矩阵的(n_v-1)行就可以完全说明所对应的图。由于一个关联矩阵的每列正好只有两个1或全为0，所以我们很容易从所给出的(n_v-1)行中去恢复那移去的一行。

从关联矩阵中移去一行的$(n_v-1)\times n_e$的子矩阵称为缩减关联矩阵，或称降阶关联矩阵A_f。也有的书上称为关联子矩阵。对应原来矩阵中移去的那一行的点叫作参考点。一个连通图的任何一点都可以作为参考点。例如，在图5—1的$A(G)$中移去行6则成为

$$A_f=\begin{array}{c} \\ 1\\2\\3\\4\\5\end{array}\begin{array}{c}\begin{matrix} a & b & c & d & e & f & g & h\end{matrix}\\ \begin{bmatrix} 0&0&0&1&0&1&0&0\\ 0&0&0&0&1&1&1&1\\ 0&0&0&0&0&0&0&1\\ 1&1&1&0&1&0&0&0\\ 0&0&1&1&0&0&1&0\end{bmatrix}\end{array} \qquad (5-36)$$

这就是一个缩减关联矩阵，点6就是参考点。参考点的编号常用0。

在关联矩阵的定义中，我们规定了没有自环的图。其实，有自环的图也可以用关联矩阵来表示它。如果在图5—1的点5上关联着一个自环k，如图5—4所示，那么它的关联矩阵应为

$$A(G)=\begin{array}{c} \\ 1\\2\\3\\4\\5\\6\end{array}\begin{array}{c}\begin{matrix} a & b & c & d & e & f & g & h & k\end{matrix}\\ \begin{bmatrix} 0&0&0&1&0&1&0&0&0\\ 0&0&0&0&1&1&1&1&0\\ 0&0&0&0&0&0&0&1&0\\ 1&1&1&0&1&0&0&0&0\\ 0&0&1&1&0&0&1&0&0\\ 1&1&0&0&0&0&0&0&0\end{bmatrix}\end{array} \qquad (5-37)$$

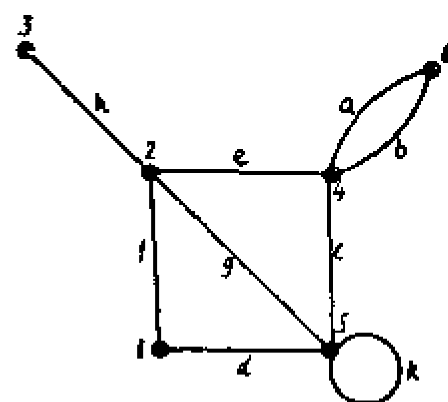


图 5—4

这里多了一列k，这列表示自环的一列都记以0。这个矩阵$A(G)$可以完全表达图5—4的各种信息，不过上述的6个信息中的1，2，3应略作修改如下：

（一）A的每列中1的和为模2零。

（二）每1行的1的数目等于对应点的次数减去自环数的双倍。

（三）都是0的一行代表一个孤立点；都是0的一列代表一个自环。

有自环的图虽然可以用一个关联矩阵来表示，如式（5—37）所示，不过，有自环的$A(G)$中表示自环的一列都是0，它不能表示出这个自环关联到哪个点上。因此，反过来讲，由关联矩阵$A(G)$不能唯一地作出一个几何图形G来。这是这种矩阵表示法的缺点。所以在关联矩阵的定义中还是规定着没有自环的。

第三节 树的矩阵

树是一个连通图的子图。在我们讲到树与关联矩阵之前，我们要先了解一个矩阵的主子

矩阵。

一个有 $p\times q$ 阶的矩阵，如果一个方子矩阵的阶数等于原矩阵中 p 和 q 中的较小数，这个子矩阵就叫主子矩阵 A_m。这个主子矩阵的行列式就叫作主行列式。

例如图 5－1 的缩减关联矩阵 A_f 的式（5－36）中包括 5 行和 8 列。既然行数小于列数，所以其中的一个主子矩阵为

$$A_m=\begin{matrix} & a & b & c & d & e \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 0 & 0 & 0 \\ 4 & 1 & 1 & 1 & 0 & 1 \\ 5 & 0 & 0 & 1 & 1 & 0 \end{matrix} \qquad (5-38)$$

为了显示树与关联矩阵之间的关系，让我们假定一个有 n_v 个点的连通图 G。设 A_f 是图 G 的一个缩减关联矩阵，既然 A_f 的秩数根据定理 5－2 等于 A_f 的行数，因此至少有一个主子矩阵的行列式值不是零的。考虑 G 的一个子图 g_m，它的那些边对应着一个非奇异主子矩阵 A_m 的那些列。因为 A_m 是非奇异的，A_m 一定是 g_m 的一个缩减关联矩阵。既然 A_m 的行数是 n_v-1，并且每一行对应着除参考点以外的一个点，g_m 有 n_v 个点。又因为 A_m 是一个方矩阵，并且每列对应着一条边，g_m 有 n_v-1 条边。而且，g_m 的秩数为 n_v-1。所以 g_m 一定是连通的。这样的一个子图就是一棵生成树。因此对生成树也可以这样来下定义，即有 n_v 个点的一个连通图 G 的一棵生成树是一个连通的子图，它有 n_v 个点和 n_v-1 条边。

现在可以明确地说，如果一个图的一个子图 g_m 是一棵生成树，那么一个各列对应着 g_m 的所有边的主子矩阵 A_m 是非奇异的。式（5－38）所示的主子矩阵是奇异的，因为这个子图的各边不是图 5－1 的生成树。

以图 5－5 为例，它的缩减关联矩阵为

$$A_f=\begin{matrix} & a & b & c & d & e \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 1 & 1 & 0 \\ 3 & 0 & 0 & 0 & 1 & 1 \end{matrix}$$

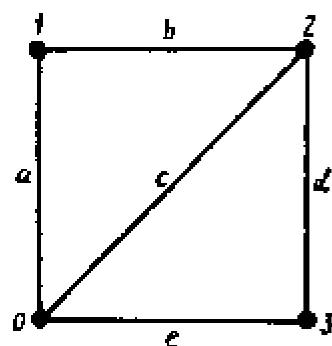


图 5－5

A_f 的非奇异子矩阵和它相应的子图即生成树如表 5－1 所示。

第四节 回路矩阵

正象一个关联矩阵显示出一个图中每条边的位置，回路矩阵则显示出每条边所属的回路。

设一个图 G 中不同回路数为 q，G 中的边数为 n_e，一个图 G 的一个回路矩阵 B 是一个 $q\times n_e$ 阶的（0，1）——矩阵，定义如下：

$$B=[b_{ij}] \qquad (5-39)$$

式中 $b_{ij}=1$，如果边 j 是在回路 i 之中；

$b_{ij}=0$，如果边 j 不在回路 i 之中。

为了着重表明 B 是图 G 的一个回路矩阵，B 也可以写成 $B(G)$。

非奇异子矩阵和生成树 表 5 — 1

非奇异子矩阵	生　成　树	非奇异子矩阵	生　成　树
$T_1\ \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{ccc} a & b & d \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{array}$		$T_5\ \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{ccc} a & d & e \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{array}$	
$T_2\ \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{ccc} a & b & e \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}$		$T_6\ \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{ccc} b & c & d \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{array}$	
$T_3\ \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{ccc} a & c & d \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{array}$		$T_7\ \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{ccc} b & c & e \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{array}$	
$T_4\ \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{ccc} a & c & e \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}$		$T_8\ \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{ccc} b & d & e \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{array}$	

图 5 — 1 中的线性图G有 4 个不同回路，即（a，b），（c，e，g），（d，f，g）和（c，d，e，f），所以它的回路矩阵$B(G)$为 4 行 8 列（0，1）——矩阵如下：

$$B(G)=\begin{array}{c} \\ C_1 \\ C_2 \\ C_3 \\ C_4 \end{array}\begin{pmatrix} a & b & c & d & e & f & g & h \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \qquad (5-40)$$

观察$B(G)$，我们得到下列信息：

（一）一列元素都是 0 对应着不属于任何回路的一条边。

（二）$B(G)$中每一行是一个回路。

（三）与关联矩阵不一样，一个回路矩阵可以代表一个自环，不过那个对应行只有一个 1。

（四）每行中 1 的数目等于对应回路中的边数。

（五）如果图G是个可分（分离）图，并且包括两个部分g_1和g_2，那么回路矩阵$B(G)$可以写成分块对角矩阵形式，如

$$B(G)=\begin{bmatrix} B(g_1) & \vdots & 0 \\ \cdots\cdots & \vdots & \cdots\cdots \\ 0 & \vdots & B(g_2) \end{bmatrix} \tag{5-41}$$

其中$B(g_1)$和$B(g_2)$分别是g_1和g_2的回路矩阵，这是因为事实上g_1中的回路与g_2中的回路没有共同边的。反过来也是这样。

（六）一个回路矩阵中任何两行或两列的置换只相当于回路和边的重新编号。

在一个图的回路矩阵中，既然$B(G)$的每一行代表着一个回路，两行或几行的模2和为一个回路。例如，在式（5—40）中，C_2和C_3这两行的模2和就是C_4。这就是说$C_2\oplus C_3=C_4$。

关于没有自环的图G的关联矩阵和回路矩阵间有一个重要关系如下。

定理5—3：设一个没有自环的图G，它的回路矩阵B和关联矩阵A中那些列的排列都是按照边的编号顺序的，那么B的每行正交于A的每行。就是

$$AB^t=BA^t=0 \qquad (模2) \tag{5-42}$$

式中上标t表示转置矩阵，O表示一个矩阵的元素值都是二进制零。即零矩阵。

证：设想图G中一个点v和一个回路C。这个点v可能在C之中，也可能不在C之中。如点v不在C之中，那么回路C中没有边关联到v。另一方面，如果点v是在C之中，那么回路C中关联到点v的边数正好是2。

有了上面的论点后，我们来注意A中的行i和B中的行j。既然各边是按顺序排列的，只有既关联到点i，也在回路j之内的那条特定边所对应的位置的元素值不是零。

如果点i不在回路j之中，就没有这种非零元素值，并且两行的点乘积是零。如果点i是在回路j之中，那么各个元素值乘积之和中正好有两个1。因为根据模2算术$1+1=0$，任意两行（一行出自A，另一行出自B）的点乘积均为零。▮

这个定理也可以用下面的证法。

设图G有n_e条边和n_v个点，所以它的关联矩阵A是

$$A=\begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_{n_v} \end{bmatrix}$$

它的回路矩阵B是

$$B=\begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_p \end{bmatrix}$$

设A中各列的排列法是使A和B中的列k都代表$k=1,2,\cdots,n_e$的同一边，那么

$$AB^t=\begin{bmatrix} A_1B_1^t & A_1B_2^t & \cdots & A_1B_p^t \\ A_2B_1^t & A_2B_2^t & \cdots & A_2B_p^t \\ \vdots & \vdots & & \vdots \\ A_{n_v}B_1^t & A_{n_v}B_2^t & \cdots & A_{n_v}B_p^t \end{bmatrix} \tag{5-43}$$

式中

$$A_i B_j' = \sum_{k=1}^{n_e} a_{ik} b_{jk} \tag{5—44}$$

注意 $a_{ik}=1$ 的意思是边 k 关联到点 i，$b_{jk}=1$ 的意思是边 k 是在 B 中行 j 所代表的回路之中。所以式（5—43）中非零项的项数就是那些边关联到点 i 并且同时又在回路 j 之中的边数。既然回路中的每一点的次数为偶数，式（5—43）中非零项的项数就是偶数。因为我们用模 2 代数，每一非零项的值为 1 。所以关于所有的 i 和 j

$$A_i B_j' = 0 \tag{5—45}$$

例 5—3　图 5—1 中图 G 的关联矩阵 $A(G)$ 见式（5—33），它的回路矩阵见式（5—40）。在这两个矩阵中，检查各边的排列顺序是完全相同的。所以可以用式（5—42）来求 A 和 B' 的乘积。

$$AB' = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{matrix} a & b & c & d & e & f & g & h \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \begin{matrix} C_1 & C_2 & C_3 & C_4 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{array}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = 0$$

凡是有自环的图 G，如果用图 5—4 的矩阵 $A(G)$ 来表示，那么式（5—42）也是适用的。

在第三章的第六节中，我们知道对于 G 的任何一棵生成树中的基本回路组是图 G 的独立回路。有了这些回路，其它的回路是可以通过环和运算来求得的。所以，在一个回路矩阵中，如果我们只保留那些相当于基本回路的行而抛弃其它行，我们还是不会损失任何信息的，因为这些抛弃的其它行是可以从基本回路组中的那些行构成的。上面已经讲过，式（5—40）中的行 4 只是行 2 和行 3 的模 2 和。

当一个回路矩阵的一个于矩阵中所有行对应于一个基本回路组，这个于矩阵叫作一个基本回路矩阵 B_f。图 5—6 是一个线性图，有 5 个点和 7 条边，其中粗线所示是它的一棵生成树。

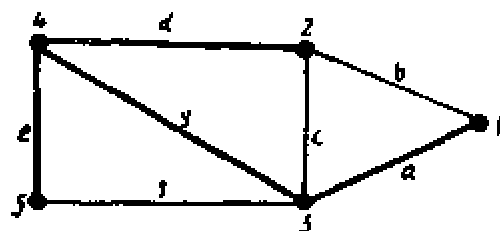


图 5—6

对应这棵生成树的基本回路矩阵为

$$B_f=\begin{array}{c} \begin{array}{ccc:cccc} b & c & f & a & d & e & g \end{array} \\ \left[\begin{array}{ccc:cccc} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{array}\right] \end{array} \qquad (5-46)$$

在矩阵A和B中，行（或列）的置换并不影响B_f，所以我们可以把式（5—40）的回路矩阵B重新排列。先把图G中生成树以外的n_e-n_v+1条弦排在矩阵的最左n_e-n_v+1列中，例如图5—6中的弦b，c，f排在最左方。再把行重新排列，使行1对应着由列1的弦所构成的基本回路，行2对应着由列2的弦所构成的基本回路，余类推。这就是式（5—46）基本回路矩阵的排列法。在矩阵B_f中，虚线以左的方矩阵是一个单位矩阵U。所以一个基本回路矩阵B_f可以写成为

$$B_f=[U_N \vdots B_t] \qquad (5-47)$$

其中U_N的阶为$n_e-n_v+1=N$，而B_t的列是对应着生成树各树枝B_t的$N\times(n_v-1)$阶子矩阵。

从式（5—45）可知B_f的秩数R为

$$R=n_e-n_v+1=N$$

既然B_f是回路矩阵B的一个子矩阵，所以B的秩数R_B为

$$R_B\geqslant n_e-n_v+1 \qquad (5-48)$$

定理5—4：如果B是一个有n_e条边和n_v个点的连通图G的一个回路矩阵，那么B的秩数R_B为

$$R_B=n_e-n_v+1 \qquad (5-49)$$

证：如果A是G的一个关联矩阵，由式（5—42）知

$$AB'=0$$

为了证明式（5—49），先介绍一个雪尔凡斯探定理。

若矩阵P是$r\times s$阶，Q是$s\times t$阶，且$PQ=0$，则

$$R_P+R_Q\leqslant S$$

式中R_P是P矩阵的秩，R_Q是Q矩阵的秩。

据此得

$$R_A+R_B\leqslant n_e$$

式中R_A为关联矩阵A的秩数，R_B为回路矩阵B的秩数。这就是说

$$R_B\leqslant n_e-R_A \qquad (5-50)$$

根据定理5—2

$$R_A=n_v-1 \qquad (5-51)$$

以式（5—51）代入式（5—50）得

$$R_B\leqslant n_e-n_v+1 \qquad (5-52)$$

但是由式（5—48），我们知道

$$R_B\geqslant n_e-n_v+1 \qquad (5-53)$$

从式（5—52）和（5—53），我们一定会知道

$$R_B=n_e-n_v+1。 \qquad \blacksquare$$

利用回路矩阵，我们可以解决开关网络中某些问题。假使有一个未说明内部联系的黑盒子，其中有8个开关a，b，c，d，e，f，g和h所组成的一个开关网络。这些开关可

以从盒子的外面开或关。现在要求不打开盒子而找出内部开关的连接法。

回答这个问题的一个方法是在盒子上已备有的两个端子上串联一个灯泡L，一个电池E，并另加一个开关k，如图5—7所示。找寻内部开关连接法的方法是测试哪些开关的组合能够使灯泡发亮。

假定我们发现闭合下列开关的组合能够使灯泡发亮：

(a, b, f, h, k)，(a, b, g, k)，(a, e, f, g, k)，(a, e, h, k)，(b, c, e, h, k)，(c, f, h, k)，(c, g, k)，(d, k)。

用图论的解法是把这个开关网络作为一个线性图G，并把它的边代表着那些开关，它的点是开关之间的连接点。我们可以假定这个线性图G是一个连通图，并且没有自环。灯泡L的发亮表征着灯泡和电池构成回路。我们可以把上面的开关组合表作为图G的部分回路。这样我们就可以得到这个图G的回路矩阵为

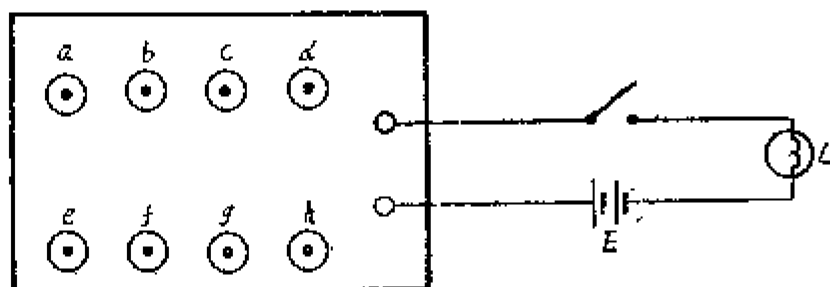


图 5—7

$$
B(G)=\begin{array}{c}\\ C_1\\ C_2\\ C_3\\ C_4\\ C_5\\ C_6\\ C_7\\ C_8\end{array}
\begin{array}{c}
\begin{array}{ccccccccc} a & b & c & d & e & f & g & h & k\end{array}\\
\left(\begin{array}{ccccccccc}
1&1&0&0&0&1&0&1&1\\
1&1&0&0&0&0&1&0&1\\
1&0&0&0&1&1&1&0&1\\
1&0&0&0&1&0&0&1&1\\
0&1&1&0&1&0&0&1&1\\
0&0&1&0&0&1&0&1&1\\
0&0&1&0&0&0&1&0&1\\
0&0&0&1&0&0&0&0&1
\end{array}\right)
\end{array}
$$

为了简化这个矩阵，我们应该除去那些明显地是多余的回路。通过下列的环和运算，我们可以得到

$$(a, b, g, k)\oplus(c, f, h, k)\oplus(c, g, k)=(a, b, f, h, k)=C_1$$

$$(a, b, g, k)\oplus(a, e, h, k)\oplus(c, g, k)=(b, c, c, h, k)=C_5$$

$$(a, e, h, k)\oplus(c, f, h, k)\oplus(c, g, k)=(a, e, f, g, k)=C_3$$

所以从$B(G)$中除去C_1，C_3和C_5后，我们并不损失什么信息。从8行中删除3行后，得一个5×9的矩阵B_1如下：

$$
B_1=\begin{array}{c}
\begin{array}{ccccccccc} a & b & c & d & e & f & g & h & k\end{array}\\
\left(\begin{array}{ccccccccc}
1&1&0&0&0&0&1&0&1\\
1&0&0&0&1&0&0&1&1\\
0&0&1&0&0&1&0&1&1\\
0&0&1&0&0&0&1&0&1\\
0&0&0&1&0&0&0&0&1
\end{array}\right)
\end{array}
$$

下一步是置换行或列，使上式成为如式（5—47）那样的形式。现在先置换上式中的几列，得

$$
B_2=\begin{array}{c} \begin{matrix} b & e & f & g & d & a & c & h & k \end{matrix} \\ \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}
$$

再将行4加到行1，得

$$
B_3=\begin{array}{c} \begin{matrix} b & e & f & g & d & a & c & h & k \end{matrix} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}=[U_6 : F]
$$

在矩阵B_3中没有多余回路，B_3是所需线性图的基本回路矩阵。既然B_3的秩数为5，并且这个网络假定是连通的，即这是一个连通图，所以我们关于这个图有下列信息：$n_e=9$，$N=5$，$R=4$，$n_n=5$。

从图的基本回路矩阵来构成一个图是困难的，但是从它的关联矩阵来构图是比较简单的。所以我们先要从B_3找出一个关联矩阵。

根据定理5—4我们知道，一个关联矩阵的每行是正交于B中的每行的，所以必须先找出一个4×9矩阵M，它的各行是线性独立的，并且正交于B_3的各行的。

因为

$$B_3=[U_5 : F]$$

所以一个正交于B_3的矩阵M是

$$M=[-F' : U_4]=[F' : U_4] \tag{5—54}$$

这是因为在模2算术中，$-1=1$。因此

$$
M=\begin{array}{c} \begin{matrix} b & e & f & g & d & a & c & h & k \end{matrix} \\ \left(\begin{array}{ccccc:cccc} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right) \end{array}
$$

很清楚，M的秩数为4，并且容易检查出

$$B_3M'=0 \tag{5—55}$$

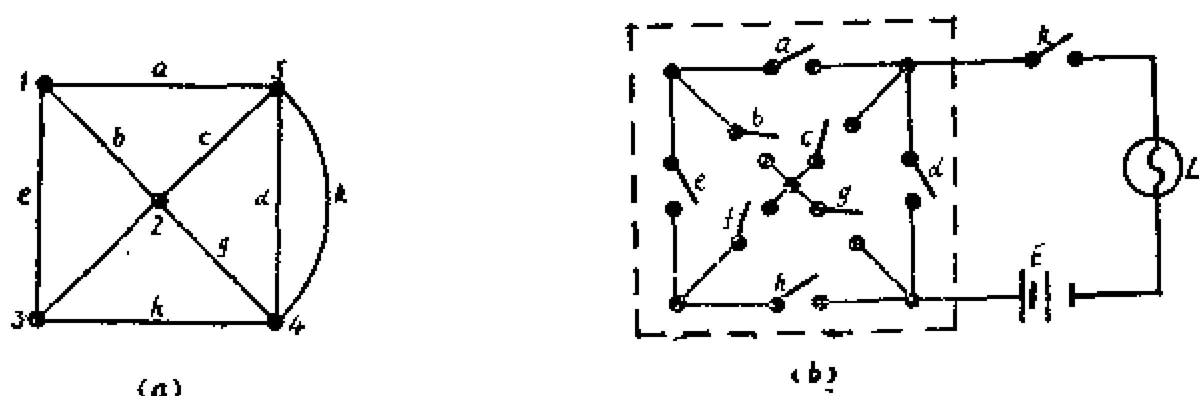
从所求得的M看来，因为每列最多超过两个1，所以这还不是一个缩减关联矩阵。这可以把M的行3加到行4，这样就得到A_f。

$$A_f=\begin{pmatrix} b & e & f & g & d & a & c & h & k \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

矩阵A_f是一个缩减关联矩阵。要得到关联矩阵A，我们可以在A_f中添加一 个 第 5 行，使得每行正好只有两个1，如

$$A=\begin{pmatrix} b & e & f & g & d & a & c & h & k \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}$$

从关联矩阵A我们可以构成所需要的线性图，如图 5 — 8（a）所示，和所需要的开关网络如图 5 — 8（b）所示。



（a）　　　　（b）

图　5—8

第五节　截割组矩阵

类似回路矩阵，我们提出截割组矩阵C，其中行对应着截割组，列对应着图的边。定义如下：

$$C=[C_{ij}] \tag{5—56}$$

式中　$C_{ij}=1$，如果边j是在截割组i之中；

　　　$C_{ij}=0$，如果边j不在截割组i之中。

图 5 — 9 是一个图和它的截割组。对应这个图的截割组矩阵为

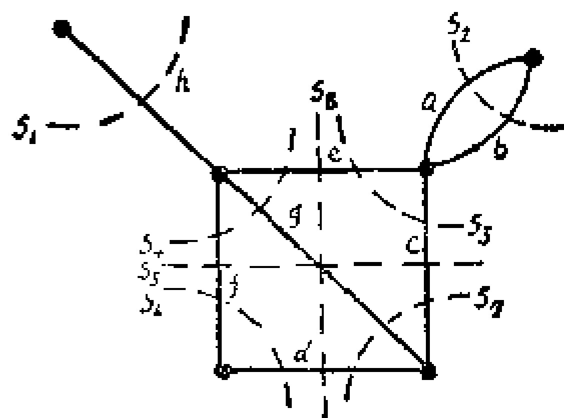


图　5—9

$$C(G)=\begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ S_8 \end{array}\begin{array}{c} a\ b\ c\ d\ e\ f\ g\ h \\ \left[\begin{array}{cccccccc} 0&0&0&0&0&0&0&1\\ 1&1&0&0&0&0&0&0\\ 0&0&1&0&1&0&0&0\\ 0&0&0&0&1&1&1&0\\ 0&0&1&0&0&1&1&0\\ 0&0&0&1&0&1&0&0\\ 0&0&1&1&0&0&1&0\\ 0&0&0&1&1&0&1&0 \end{array}\right]\end{array} \qquad (5-57)$$

从截割组矩阵$C(G)$可以得到下列信息：

（一）象在关联矩阵中那样，在一个截割组矩阵中置换行或列就相当于分别把截割组或边重新编号。

（二）$C(G)$中每一行是一个截割组。

（三）一个自环的一列都是0。

（四）在截割组矩阵中，平行边产生恒等列〔例如式（5—57）中的列1和列2〕。

（五）在一个不可分图中，所有关联到一点的每个边组是一个截割组。所以在关联矩阵$A(G)$中的每一行也包括在截割组矩阵$C(G)$中作为一行。换句话讲，一个不可分图G中的$C(G)$包含着$A(G)$。对于一个可分图呢，每个部分的关联矩阵含于截割组矩阵之中。例如，在图5—9中部分（c，d，e，f，g）的关联矩阵是从式（5—57）中删除列a，b和h，以及行S_1，S_2，S_5和S_8后所余留下来的$C(G)$的4×5子矩阵。

（六）由上一条（即5）看来，如R_C为矩阵C的秩数，R_A为矩阵A的秩数，那么

$$R_C\geqslant R_A$$

所以，对于一个有n_v个点的连通图来说

$$R_C\geqslant n_v-1 \qquad (5-58)$$

（七）既然一个截割组和一个回路所共有的边数总是偶数，倘然在B和C中各边用同样顺序排列的话，那么C中每一行是正交于B中每一行的，即

$$BC^t=CB^t=0\quad(\text{模}2) \qquad (5-59)$$

对式（5—59）应用雪尔凡斯探定理，则

$$R_B+R_C\leqslant n_e \qquad (5-60)$$

又因为对一个连通图来说

$$R_B=n_e-n_v+1 \qquad (5-61)$$

$$R_C\leqslant n_v-1 \qquad (5-62)$$

由式（5—58）和（5—62）可见

$$R_C=n_v-1 \qquad (5-63)$$

由此得连通图G的一个重要定理。

定理5—5：截割组矩阵$C(G)$的秩数等于关联矩阵$A(G)$的秩数，后者等于图G的秩数。

象在回路矩阵中一样，截割组矩阵中一般也有许多冗余的（或线性相关的）行。所以，除截割组矩阵外，也有一个基本截割组矩阵，它的定义如下：

一个有n_e条边和n_v个点的连通图G的基本截割组矩阵C_f是截割组矩阵C的一个$(n_v-1)\times n_e$子矩阵，它的那些行对应着图G的某一棵生成树的一组基本截割组。

一个基本截割组矩阵C_f可以分块成两个子矩阵，一个是（n_v-1）阶的单位矩阵U_{n_v-1}，一个是对应着弦的C_c，如

$$C_f=[C_c \vdots U_{n_v-1}] \qquad (5-64)$$

其中C_f的后n_v-1列组成对应着生成树的n_v-1个枝的单位矩阵，前n_e-n_v+1列组成对应着弦的C_c。图5—10是一个连通图G，粗线所示是一棵生成树。对应这棵生成树的基本截割组矩阵为

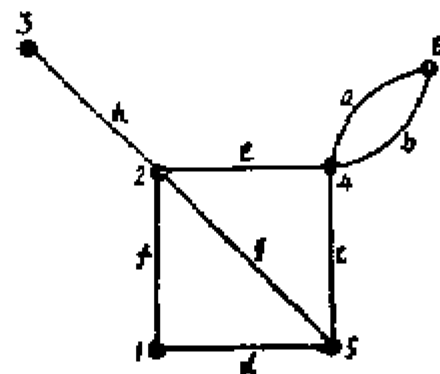


图 5—10

$$C_f=\begin{array}{c} \begin{matrix} b & c & d & & a & e & f & g & h \end{matrix} \\ \left(\begin{array}{ccc:ccccc} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right) \end{array}=[C_c \vdots U_5] \qquad (5-65)$$

比较式（5—57）和式（5—65），我们可以看出基本截割组矩阵C_f是由式（5—57）中的S_2，S_3，S_6，S_7和S_1所组成的。这些截割组就是图5—10中粗线所示的生成树的基本截割组。因为$C(G)$中的S_4，S_5和S_8 3行是冗余的截割组，所以删除了。这些截割组可以由基本截割组通过环和运算来得到的，即

$$S_4=S_5 \oplus S_6 \oplus S_7$$
$$S_5=S_6 \oplus S_7$$
$$S_8=S_3 \oplus S_7$$

在讲到截割组矩阵时，我们只讲连通图。对于分离图呢，我们就要一个部分一个部分地处理。

第六节 路 径 矩 阵

路径矩阵P是对图G中一对特定点而言的。譬如说是对点（x，y）而言的，它的表达式为

$$P(x, y)=[p_{ij}] \qquad (5-66)$$

式中 $p_{ij}=1$，如果边j是在路径i之中；

$p_{ij}=0$，如果边j不在路径i之中。

以图5—1中的线性图G为例，在点3和点4之间有3条不同的路径，即（h，e），（h，g，c）和（h，f，d，c）。这样我们得到点3和4之间的3×9路径矩阵如下：

$$P_{(3,4)}=\begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \end{array}\begin{array}{c} \begin{matrix} a & b & c & d & e & f & g & h \end{matrix} \\ \left(\begin{matrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{matrix}\right) \end{array} \qquad (5-67)$$

从一个图G的一个路径矩阵$P(x, y)$中，我们得到下列信息：

（一）都是0的一列对应着那条边不在点x和y之间的任何路径之中。

（二）都是1的一列对应着那条边是在点x和y之间的任何路径之中。

（三）没有一行都是0的。

（四）在$P(x, y)$中任何两行的环和对应着一个回路或者回路的一个边不共用并集。

定理5—6：如果一个连通图的各边在关联矩阵A中各列的排列顺序与路径矩阵$P(x, y)$中的顺序相同的话，那么

$$AP'_{(x,y)}=M \text{（模 2）} \tag{5—68}$$

式中矩阵M的x和y两行都是1，而其余的n_v-2行都是0。

例5—4　以图5—1的关联矩阵A乘以转置了的上述$P_{(3,4)}$

$$AP'_{(3,4)}=\begin{pmatrix}0&0&0&1&0&1&0&0\\0&0&0&0&1&1&1&1\\0&0&0&0&0&0&0&1\\1&1&1&0&1&0&0&0\\0&0&1&1&0&0&1&0\\1&1&0&0&0&0&0&0\end{pmatrix}\begin{pmatrix}0&0&0\\0&0&0\\0&1&1\\0&0&1\\1&0&0\\0&0&1\\0&1&0\\1&1&1\end{pmatrix}=\begin{matrix}1\\2\\3\\4\\5\\6\end{matrix}\begin{pmatrix}0&0&0\\0&0&0\\1&1&1\\1&1&1\\0&0&0\\0&0&0\end{pmatrix}$$

一般地讲，一个路径矩阵中所包含的关于图的信息要比上面讲过的关联矩阵，回路矩阵和截割组矩阵为少。

第七节　邻 接 矩 阵

代表一个图G的关联矩阵，有时还不如用邻接矩阵来得方便。有n_v个点和没有平行边的图G的邻接矩阵是$n_v \times n_v$对称二进制矩阵。它的表达式为

$$X=[x_{ij}] \tag{5—69}$$

式中　$x_{ij}=1$，如果在点i和j之间有一条边；

$x_{ij}=0$，如果在点i和j之间没有边。

图5—11是一个简单的连通图G。它的邻接矩阵为

$$X=\begin{matrix} \\ 1\\2\\3\\4\\5\\6\end{matrix}\begin{matrix}1&2&3&4&5&6\\ \begin{pmatrix}0&1&0&0&1&1\\1&0&0&1&1&0\\0&0&0&1&0&0\\0&1&1&0&1&1\\1&1&0&1&0&0\\1&0&0&1&0&0\end{pmatrix}\end{matrix} \tag{5—70}$$

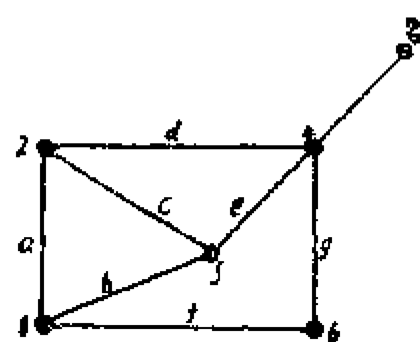


图　5—11

从图G的邻接矩阵X中可以得到下列信息：

（一）当且仅当图G没有自环时，X的主对角线的元素值都是0，点i有一个自环时，对应着$x_{ii}=1$。

（二）邻接矩阵的定义规定着没有平行边。

（三）如果图G没有自环并且当然没有平行边，一个点的次数等于X的对应行或列中的1的数目。

（四）行和对应列的置换意味着点的重新编号。可是必须注意的是行和列必须用同样顺

序排列。所以，如果在X中两行互换，那么相应的两列也必须互换。因此当且仅当两个没有平行边的图G_1和G_2的邻接矩阵$X(G_1)$和$X(G_2)$有下列关系

$$X(G_2)=R^{-1}X(G_1)R \tag{5—71}$$

（式中R是置换矩阵）时，G_1和G_2是同构的。

（五）当且仅当一个图G的邻接矩阵$X(G)$可以分块为

$$X(G)=\left(\begin{array}{c|c} X(g_1) & 0 \\ \hline 0 & X(g_2) \end{array}\right) \tag{5—72}$$

其中$X(g_1)$是部分g_1的邻接矩阵，$X(g_2)$是部分g_2的邻接矩阵时，图G是分离的，并且有两个部分g_1和g_2。这种分块清楚地意味着没有边把子图g_1中任何一点连通到子图g_2的任何一点。

（六）给定任何一个n_c阶的正方的、对称的二进制矩阵Q，人们总是能够构成一个n_c个点（而无平行边）的图G，使Q就是G的邻接矩阵X。

邻接矩阵的定义中虽然规定图是没有平行边的，其实，有平行边的图也可以用邻接矩阵来代表，不过，平行边的x_{ij}不等于1而等于平行边的边数。这样一来，这个邻接矩阵不再是二进制矩阵，即不再是(0，1)-矩阵。这种矩阵运算或应用起来有时是不方便的，所以在定义邻接矩阵时，还是规定着没有平行边。

要了解邻接矩阵X的幂，我们可以图5—11的简单图G的6×6邻接矩阵X，即式（5—70）自己相乘，结果得到另外一个6×6对称矩阵X^2如下所示。注意这是整数环的普通矩阵乘法，而非模2算术。

$$X^2=\begin{pmatrix} 3 & 1 & 0 & 3 & 1 & 0 \\ 1 & 3 & 1 & 1 & 2 & 2 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 3 & 1 & 0 & 4 & 1 & 0 \\ 1 & 2 & 1 & 1 & 3 & 2 \\ 0 & 2 & 1 & 0 & 2 & 2 \end{pmatrix}$$

在X^2中一个非对角线上的元素值X^2_{ij}，即X^2中的$i-j$处的元素值＝X的行i和列j（或行j）点乘积中的1的数目。

$X^2_{ij}=X$的行i和行j都有1的位置数

＝既和点i邻接又和点j邻接的点数

＝在点i和j间有两段间隔的不同路径数

同样，在X^2中对角线上i的元素值是矩阵X中行（或列）i的1的数目。所以，如果G没有自环的话，X^2中对角元素值等于对应点的次数。

矩阵的幂适用交换律，即

$$XX^2=X^2X=X^3 \tag{5—73}$$

既然两个正方对称矩阵的乘积是一个对称矩阵，X^3也是一个对称矩阵。这里讲的乘法是通常的乘法而非模2乘法。

图5—11中简单图G的X^3为

$$X^3=\begin{pmatrix}2&7&3&2&7&6\\7&4&1&8&5&2\\3&1&0&4&1&0\\2&8&4&2&8&7\\7&5&1&8&4&2\\6&2&0&7&2&0\end{pmatrix}$$

在了解这个矩阵中元素值的意义之前，先介绍一个术语边序列。所谓边序列是若干条边的连续序列，其中每条边（当然除去第一条边和最后一条边）的一个端点与前一条边所共有，而另一个端点与后一条边所共有。第二章中所讲到的行走，路径和回路都是边序列的特例。在一个边序列中，一条边可能出现不止一次。

现在让我们看看上面的矩阵X^3的i-j处的元素值

$X^3_{ij}=X^2$的行i与X的列（或行）j的点乘积

$=\sum_{k=1}^{n}X^2$的i-k处的元素值与X的k-j处的元素值的点乘积

$=\sum_{k=1}^{n}$从点i通过点k到点j的 3 条边的所有不同边序列数

$=$在点i与点j间 3 条边的不同边序列数

例 5 — 5　在图 5 —11中图G的X^3的1-5处的元素值是如何得来的呢？根据上面所讲，X^3中1-5处的元素值是X^2的行 1 与X的行 5 的点乘积，即

X^3中1-5处的元素值

$X^3_{15}=(3,1,0,3,1,0)\cdot(1,1,0,1,0,0)$

$=3+1+0+3+0+0=7$

在点 1 与点 5 之间 3 条边的 7 个不同边序列是 $(a,\ a,\ b)$，$(b,\ b,\ b)$，$(f,f,\ b)$，$(b,\ c,\ c)$，$(f,\ g,\ e)$，$(b,\ e,\ e)$，$(a,\ d,\ e)$。

很明显，这些边序列包括在点 1 与点 5 之间有 3 段间隔的所有路径，即 $(f,\ g,e)$ 和 $(a,\ d,\ e)$。

下一个定理概括着上面所讨论到的X，X^2和X^3的性质。

定理 5 — 7：设X是一个简单图G的邻接矩阵，那么在X^r的i-j处的元素值是点v_i与点v_j之间r条边的不同边序列数。

因为我们从上面已经知道X，X^2和X^3的ij处的元素值，所以定理 5 — 7 就可以用归纳法来证明。这是面为这个定理既然对X的$r-1$幂是正确的，那么

$$X^r=X^{r-1}\cdot X \qquad (5—74)$$

正像式（5 —73）中的X^3一样。这样论证下去就可以得到这个定理的证明。

由此得到两个推论：

（一）在一个连通图中，当且仅当X^d中的i-j处的元素值不是零而且d是最小整数时，两个点v_i与v_j（$i\neq j$）之间的距离为d。

（二）如果X是一个有n_o个点的图的邻接矩阵和

$Y=X+X^2+X^3+\cdots\cdots+X^{n_o-1}$（属于整数环）

当且仅当矩阵Y中至少有一个元素值是零时，G是分离的。

这个图G的邻接矩阵X的主对角线的元素值0代以对应于i处的点i的次数$d(i)$，而把$i \neq j$处的元素值标以$-k$，其中k为图G中点i与点j之间的边数，则这个矩阵称为G的次数矩阵D。下面是一个计算图G的所有生成树的定理，称为矩阵-树定理。

定理5—8：一个没有自环的无向标记图G中所有生成树的数目等于从这个图的次数矩阵D中删除任何一行及其对应的列而得到的余子式的值。

例5—6　图5—12的次数矩阵D为

$$D=\begin{matrix} & 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & -1 & -1 & 0 & 0 \\ 2 & -1 & 3 & -1 & -1 & 0 \\ 3 & -1 & -1 & 4 & -1 & -1 \\ 4 & 0 & -1 & -1 & 3 & -1 \\ 5 & 0 & 0 & -1 & -1 & 2 \end{matrix}$$

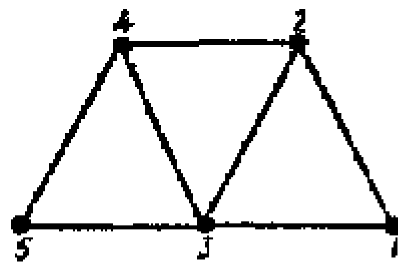


图　5—12

在D中删除行1和列1得一余子式如下

$$\begin{vmatrix} 3 & -1 & -1 & 0 \\ -1 & 4 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ 0 & -1 & -1 & 2 \end{vmatrix}$$

这个余子式的值为21。这就是图5—12的所有生成树数。

第八节　关联矩阵、回路矩阵、截割组矩阵、邻接矩阵间的关系

在一个连通图G中的基本回路矩阵B_f和基本截割组矩阵C_f是

$$B_f=[U_N \vdots B_t] \qquad (5—75)$$

$$C_f=[C_c \vdots U_{n_v-1}] \qquad (5—76)$$

其中下标t表示对应着一棵生成树各枝的子矩阵，下标c表示对应着各弦的子矩阵。

在式（5—75）和（5—76）中，生成树T是同一棵树，并设两个方程中各边的排列顺序是相同的，而且，在$(n_v-1)\times n_e$缩减关联矩阵A_f中，各边（就是各列）的排列顺序与在B_f和C_f中的相同。把A_f分块成两个子矩阵

$$A_f=[A_c \vdots A_t] \qquad (5—77)$$

其中A_t包括生成树T中对应着各枝的n_v-1个列，A_c对应着n_e-n_v+1条弦的所余留下来的子矩阵。

既然A_f和B_f中的各列是按着同一顺序排列的，由式（5—42）我们有

$$A_f B_f^T=0$$

这就是

$$[A_c \vdots A_t]\begin{bmatrix} U_N \\ \cdots \\ B_t^T \end{bmatrix}=0$$

即
$$A_c+A_t B_t^T=0 \qquad (5—78)$$

因为A_t是非奇异的，所以它的逆A_t^{-1}是存在的。在式（5—78）的两边乘以A_t^{-1}得

$$A_t^{-1}A_c=-B_t'$$

因为在模 2 算术中，$-1=1$，所以

$$B_t'=A_t^{-1}A_c \qquad (5-79)$$

同样，因为B_f和C_f中的各列是按照同一顺序排列的，根据式（5—59）我们有

$$C_fB_f'=0$$

这就是

$$[C_c \vdots U_{n_v-1}]\begin{bmatrix} U_N \\ \cdots \\ B_t' \end{bmatrix}=0$$

即

$$C_c=-B_t'=B_t' \qquad (5-80)$$

以式（5—79）代入（5—80），得

$$C_c=A_t^{-1}A_c \qquad (5-81)$$

例5—7　在图5—1，5—9和5—10中所用的图G的 3 个不同矩阵中，用（a，e，f，g，h）作为生成树，如图 5—10中粗线所示，并删除式（5—33）中的行 6 以得到 A_f，则

$$A_f=\begin{array}{c} \begin{array}{ccccccccc} b & c & d & & a & e & f & g & h \end{array} \\ \left[\begin{array}{ccc:ccccc} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{array}\right] \end{array}=[A_c \vdots A_t]$$

$$B_f=\begin{array}{c} \begin{array}{ccccccccc} b & c & d & & a & e & f & g & h \end{array} \\ \left[\begin{array}{ccc:ccccc} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \end{array}\right] \end{array}=[U_3 \vdots B_t]$$

$$C_f=\begin{array}{c} \begin{array}{ccccccccc} b & c & d & & a & e & f & g & h \end{array} \\ \left[\begin{array}{ccc:ccccc} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \end{array}=[C_c \vdots U_5]$$

从上面B_f和C_f两个矩阵中，我们立刻可以看出

$$B_t'=C_c \qquad (5-82)$$

所以我们容易验证出

$$A_t^{-1}A_c=B_t' \qquad (5-83)$$

这样得到A_f，B_f和C_f关系的 3 个结论：

（一）给出$A(G)$或A_f后，从一棵任意的生成树和它在A_f中的子图A_t着手，我们能够容易地得到B_f和C_f。

（二）给出B_f后，我们能够得到C_f；给出C_f后，我们能够得到B_f。

（三）给出B_f或C_f后，A_f一般是不能完全确定的。因为$B_t^{\prime}=C_c$，以它代入式（5—76），得

$$C_f=[B_t^{\prime} \vdots U_{n_v-1}] \qquad (5-84)$$

同样，因为$B_t=C_c^{\prime}$，以它代入式（5—75），得

$$B_f=[U_\mu \vdots C_c^{\prime}] \qquad (5-85)$$

这样，我们也得到一个基本回路矩阵B_f和一个基本截割组矩阵C_f之间的关系。

关于邻接矩阵$X(G)$，它与关联矩阵$A(G)$之间也有一定的关系。如果一个图G没有自环，它的关联矩阵$A(G)$包含着关于G的所有信息。同样，如果G没有平行边，它的邻接矩阵$X(G)$也包含着关于G的所有信息。所以，如果一个图G是一个简单图，即G既无自环，又无平行边，那么$A(G)$和$X(G)$两者就包含着G的全部信息。这样，在一个既无自环，又无平行边的简单图G中，$A(G)$和$X(G)$之间就有一个关系，它们可以从其中一个来求得另一个。

习　题

5—1　写出图4—4（b）中的简单标记图的关联矩阵。

5—2　考虑图4—6的图，对于生成树{b，c，e，h，l}以式$A_f=[A_c \vdots A_t]$，$B_f=[U_\mu \vdots B_t]$，$C_f=[C_c \vdots U_t]$的形式分别写出矩阵A_f、B_f、C_f。由实际计算证明等式$B_t^{\prime}=A_t^{-1}A_c$和$C_c=B_t^{\prime}$。

5—3　证明对于一个有k个部分、n_v个点和n_e条边的简单分离图，矩阵A、B和C的秩分别是n_v-k，n_e-n_v+k和n_v-k。

5—4　一个含有7个开关（1，2，3，4，5，6，7）的开关网络的黑盒子服从于图5—7的实验。当下列每一组开关组合与开关k被接通时，灯泡点亮。画出开关网络的连接图。开关组合为：（1，4，5），（1，4，6，7），（2，5，7），（2，6），（3，5），（3，6，7）。

5—5　如果下面是一个图G的所有生成树的表示式，请决定图G

{a，c，d，e}，{a，c，d，f}，{b，c，d，e}，{b，c，d，f}，

{a，c，e，f}，{b，c，e，f}，{a，d，e，f}，{b，d，e，f}，

{a，b，d，e}，{a，b，d，f}，{a，b，e，f}。

5—6　表明具有n_v个点的完全图中矩阵A_f，B_f，C_f和X的特性。

5—7　证明

$$B_f=[U_\mu \vdots A_c^{\prime}(A_t^{-1})^{\prime}]$$

$$C_f=A_t^{-1}A_f$$

5—8　同回路矩阵或截割组矩阵类似，对一连通图定义一个生成树矩阵，指出它的一些特性。

5—9　画出一个线性图，它的基本回路矩阵是

$$\begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & & 6 & 7 & 8 & 9 \end{matrix} \\ \left(\begin{array}{ccccc:cccc} 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right) \end{array}$$

5—10 画出一个线性图，它的基本截割组矩阵是

$$\begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & & 7 & 8 & 9 & 10 \end{matrix} \\ \left(\begin{array}{cccccc:cccc} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right) \end{array}$$

第六章　平面图和对偶图

第一节　抽象图和几何图

在第一章开头时我们提到过，一个线性图是一个抽象的事物或对象的组或集，并没有把它们在三维欧几里得空间是否可以用几何形式画出来相联系。例如一个抽象图或称组合图G可以定义为

$$G=(V,\ E,\ \Psi) \tag{6—1}$$

其中组V，譬如说，包括5个对象，称它们为1，2，3，4，5，即

$$V=(1,\ 2,\ 3,\ 4,\ 5) \tag{6—2}$$

组E，譬如说，包括7个对象，这些对象中没有一个是在组V之中的，称它们为a，b，c，d，e，f，g，即

$$E=(a,\ b,\ c,\ d,\ e,\ f,\ g) \tag{6—3}$$

这两组之间的关系由对应关系或称映射Ψ来规定。譬如说，

$$\Psi=\left\{\begin{array}{l} a\rightarrow(1,\ 3)\\ b\rightarrow(3,\ 4)\\ c\rightarrow(1,\ 4)\\ d\rightarrow(1,\ 2)\\ e\rightarrow(2,\ 4)\\ f\rightarrow(4,\ 5)\\ g\rightarrow(2,\ 5)\end{array}\right. \tag{6—4}$$

这里，符号$a\rightarrow(1,\ 3)$的意思是说，由组E中来的对象a对应着或映射到组V中来的两个对象1和3。

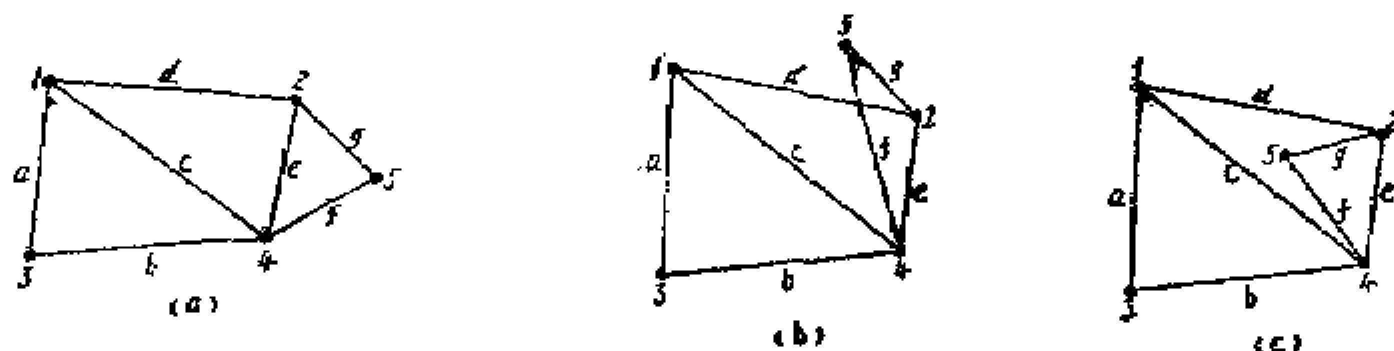


图 6—1

这个组合的抽象事物G也可以用几何图形来代表。图6—1（a）是这个图G的一种几何表示法，但是这并不是图G的本身。譬如说，我们可以把点5放在任何地方，如象图（b）和（c）就是另外两种表示法。同样，我们也可以把点1，2，3或4放在任何其它地方，或者把边a，b，c，…等画成不同形式，就能变成无数的几何形式，都能表示组V

与组E的对应关系。可是，只要不会引起混乱或误解，我们始终把图G的图形看作是图G的本身。

在前面几章中，我们不计较抽象图与几何图的区分，而在这一章中，我们就必须要注意它们的区分了。

第二节 1-同构和2-同构

一个可分图包括两个或更多个不可分子图，每一个不可分子图称为片或部分。例如，图6—2有5个片，每个片用虚线围起来，它们是由3个截割点1，2，3所划分的。一个不可分的连接图只有一个片。

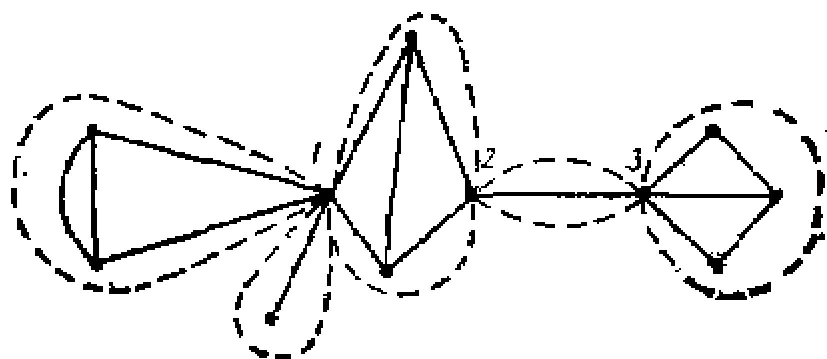


图 6—2

试观察图6—3中的分离图与图6—2可分图。因为两者各自的点数不同，它们当然不是同构，但是两者中对应的片是同构的，它们之间是有联系的。这种图叫作1-同构。1-同构的定义是：

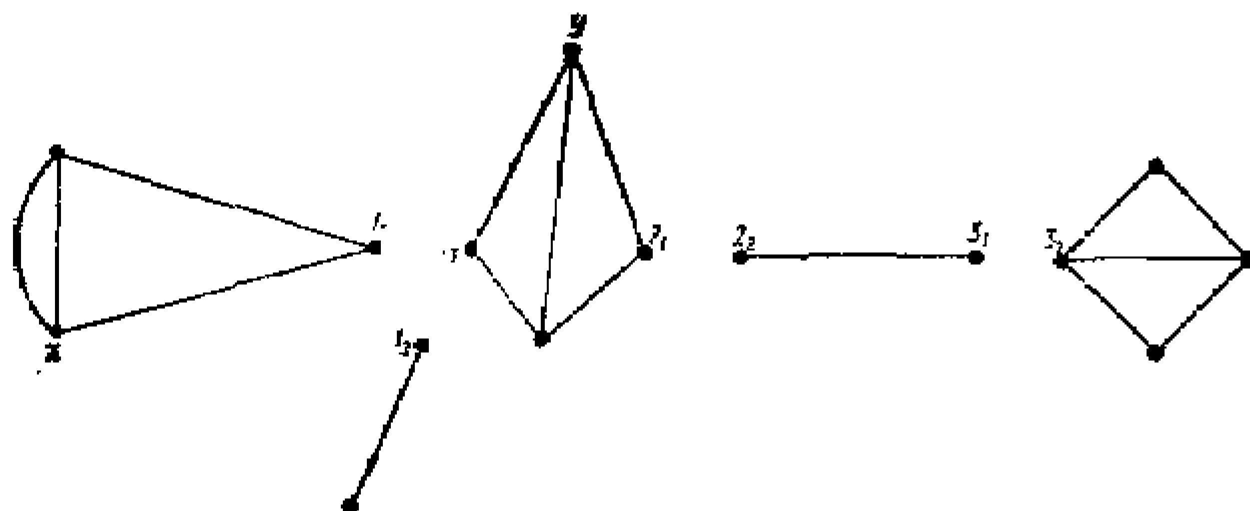


图 6—3

如果两个图G_1和G_2对各截割点重复应用下述操作Ⅰ后相互成为同构，G_1和G_2就叫作是1-同构。

操作Ⅰ：把一个截割点剖分为两个点来作成两个边不共用的子图。

由此定义可见，当且仅当两个不可分图是同构的，它们也是1-同构的。

定理6—1：如果G_1和G_2是两个1-同构图的话，G_1的秩数等于G_2的秩数，G_1的圈数等于G_2的圈数。

证：通过操作Ⅰ，每次图G中的一个截割点剖分为两个点时，G的部分数K总是增加一的。所以G的秩数R是G的点数n_v减去G的部分数k，即$R=n_v-k$，仍是不变。

两样，因为操作Ⅰ不破坏一条边，也不产生一条新边，两个1-同构图有相同的边数n_e。有相同的秩数R和相同边数n_e的两个图一定有相同的圈数N，这是因为$N=n_e-R$。∎

把图6—3中两个片中的点x和y合并在一起就成图6—4。图6—4是图6—3的1-同构。既然图6—4的片是图6—2中片的同构，这两个图也是1-同构的。所以图6—2，6—3和6—4中的3个图是互为1-同构的。

既然一个不可分图只有一个片，所以不可分图的1-同构是与同构一样的。可是，对可分图（即图的点连通度为一的）来说，1-同构是与同构不同的。同构的图也是1-同构的，但是

1-同构的图却可能不一定是同构的。这个广义同构对研究可分图是十分有用的。

那么什么叫2-同构呢？2-同构的定义为：

如果两个图G_1和G_2的回路一一对应，而且各回路之间的边又有一一对应关系，G_1和G_2叫作互为2-同构的。

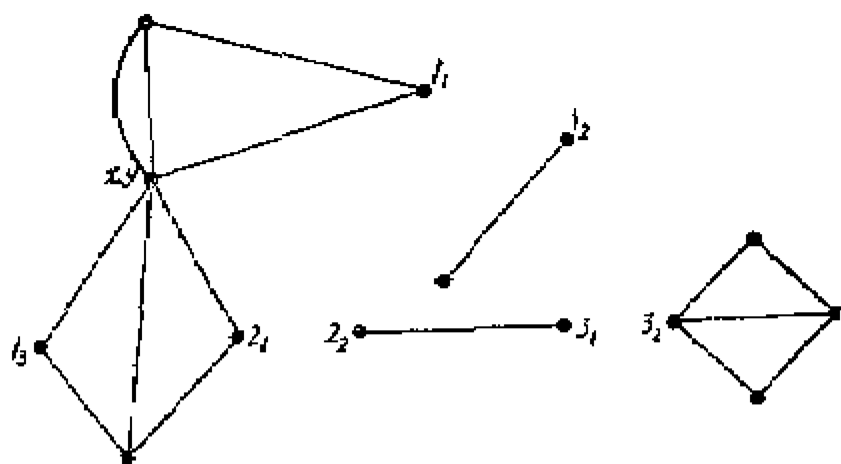


图 6－4

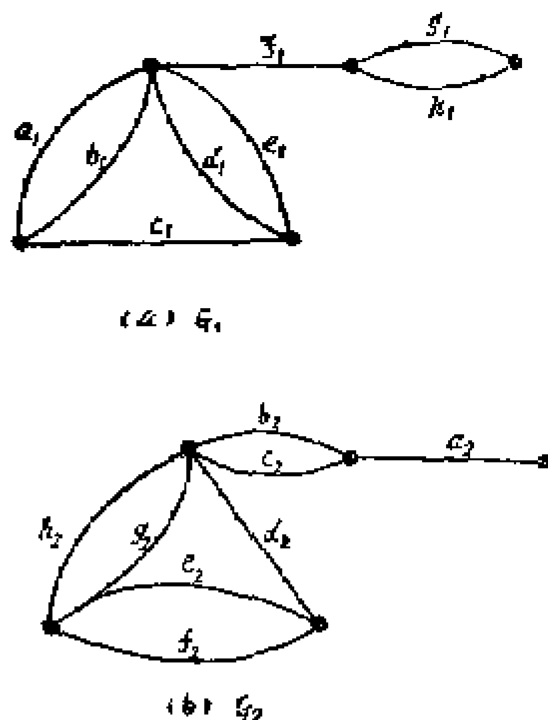


图 6－5

例如，图6－5中的两个图G_1和G_2是互为2-同构的，因为它们各边之间有一一对应关系如表6－1所示，并且它们的回路之间也有一一对应关系如表6－2所示。

边的对应表　　表6－1

G_1中的边	G_2中的边
a_1	h_2
b_1	g_2
c_1	d_2
d_1	e_2
e_1	f_2
f_1	c_2
g_1	b_2
h_1	a_2

回路的对应表　　表6－2

回路编号	G_1中的回路	G_2中的回路
c_1	a_1b_1	g_2h_2
c_2	d_1e_1	e_2f_2
c_3	g_1h_1	b_2c_2
c_4	$a_1c_1d_1$	$d_2e_2h_2$
c_5	$a_1c_1e_1$	$d_2f_2h_2$
c_6	$b_1c_1d_1$	$d_2e_2g_2$
c_7	$b_1c_1e_1$	$d_2f_2g_2$

若G_1和G_2是2-同构的，则只要把一个图的回路矩阵的行和列加以置换后，就应该与另一图的回路矩阵完全相同。从表6－2中可以看出，这7个回路中，基本回路只有前面的4个，而后面的3个是可以通过基本回路的环和运算得到的，如

$$C_2 \oplus C_4 = C_5$$

$$C_1 \oplus C_4 = C_6$$

$$C_1 \oplus C_2 \oplus C_4 = C_7$$

现在列出G_1的基本回路矩阵B_{f1}为

$$B_{f_1}=\begin{array}{c} \\ C_1 \\ C_2 \\ C_3 \\ C_4 \end{array}\begin{array}{cccccccccc} b_1 & e_1 & g_1 & c_1 & & a_1 & d_1 & f_1 & h_1 \\ 1 & 0 & 0 & 0 & \vdots & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \vdots & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & \vdots & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & \vdots & 1 & 1 & 0 & 0 \end{array}$$

把G_1的这个矩阵的各列换成G_2中对应的各列，并把各边的下标1换成2，即得G_2的基本回路矩阵B_{f_2}:

$$B_{f_2}=\begin{array}{c} \\ C_1 \\ C_2 \\ C_3 \\ C_4 \end{array}\begin{array}{cccccccccc} g_2 & f_1 & b_2 & d_2 & & h_2 & e_2 & a_2 & c_2 \\ 1 & 0 & 0 & 0 & \vdots & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \vdots & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & \vdots & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & \vdots & 1 & 1 & 0 & 0 \end{array}$$

它是与B_{f_1}完全相同的。所以G_1和G_2是互为2-同构的。由此得一定理。

定理6—2：当且仅当两个或更多个图有相同的基本回路矩阵时，它们是互为2-两构的。

根据式（5—80）或式（5—82），我们从上面这个定理又可以得到：

定理6—3：当且仅当两个或更多个图有相同的基本截割组矩阵时，它们是互为2-同构的。

上面这两个定理在测试两个图是否互为2-同构时也许不实用，可是它们是重要的，在第十章中设计线性图时会有帮助的。

下面是一种测试2-同构的几何方法。如果图G_1和G_2是互为2-同构的，那么在重复经过上面已讲过的操作Ⅰ或下面的操作Ⅱ或连续用这两种操作后，它们变成同构。

在一个2-连通图（即图的点连通度为2）G中，设点x和y是一对点，把它们从G中移去使剩余的G成为分离图。换句话说，G包括一个子图g_1和它的余图$\overline{g_1}$，这样就是g_1和$\overline{g_1}$正好有两个共同点x和y。假使我们对G进行下面的操作Ⅱ（操作后，G当然不再保留原来的图形了）。

操作Ⅱ：把点x部分为x_1和x_2，把点y部分为y_1和y_2，这样图G就划分为g_1和$\overline{g_1}$，设g中有x_1和y_1，$\overline{g_1}$中有x_2和y_2。现在把x_1与y_2以及x_2与y_1合并在一起使g_1和$\overline{g_1}$重新连接起来。很清楚，G中过去端点为x和y的那些边会随g_1或$\overline{g_1}$面移动，但不致影响最后的图形。

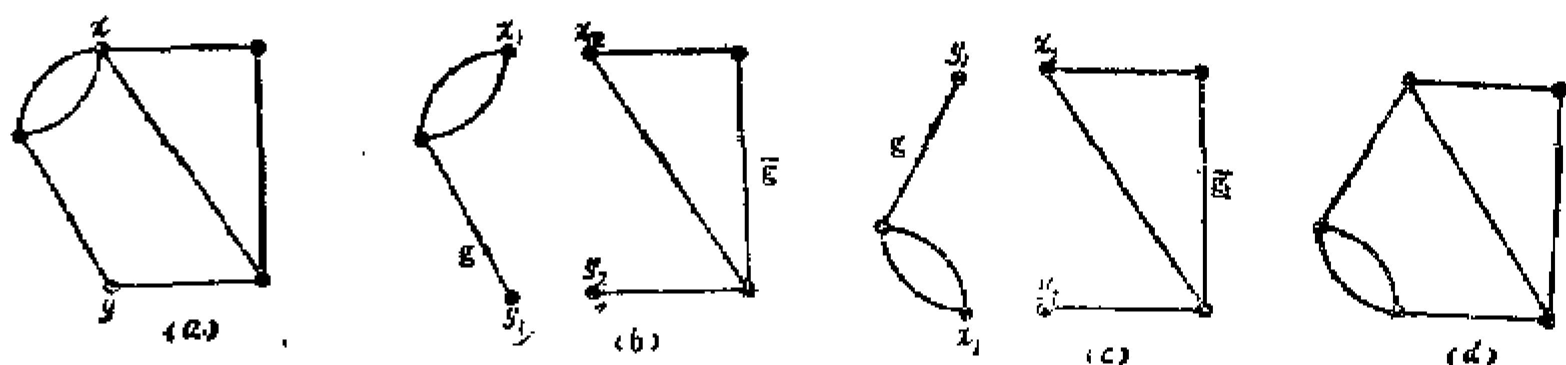


图 6—6

图6—6表现出图（a）和图（d）如何是2-同构的。值得注意的是在图（a）中，点x的次数为4，但在图（d）中，却没有一个点的次数超过3了。

在进行操作Ⅱ时，一个图的点数和边数都不减少或增多。所以在操作Ⅱ后，一个图的秩数和圈数还是不变的。同样，通过操作Ⅰ后，一个图的秩数和圈数也是不变的。所以2-同构图的秩数和圈数都是一样的。

如果两个图G_1和G_2符合下列条件，叫作有回路对应，即G_1和G_2的边之间是一一对应的，以及G_1和G_2的回路之间也是一一对应的。这样在G_1中由某些边所组成的回路就有一个在G_2中由对应边所组成的对应回路，并且反过来也是一样。很明显，同构图有回路对应的特性。

在一个可分图G中，既然每一回路局限于一个片中，在G进行操作Ⅰ时，G中每一回路还保有它的边。所以1-同构图有回路对应。

同样，让我们考虑一个图G中的一个回路进行操作Ⅱ后的情形。一个图G的回路C可能属于下面3种类型的一种：

（一）C的各边都在g_1之中；

（二）C的各边都在$\overline{g_1}$之中；

（三）C的各边有些在g_1之中，而另一些在$\overline{g_1}$之中。在这种情况时，C一定包括点x和y。

在第一和第二种情形时，操作Ⅱ不会影响C。在第三种情形时，C仍旧有原来的各边，只是在g_1中点x和y之间的那段路径（回路C的一部分）被扭转了。所以一个图G中的每一回路通过操作Ⅱ后仍保有它原来的各边。因此，2-同构图也有回路对应。

定理6－4：当且仅当两个图有回路对应时，它们是2-同构的。

另有一种测试2-同构的方法是用一个图的一个基本回路矩阵和另一个图的一个基本截割组矩阵。

设G_1的一个基本回路矩阵B_{f1}为

$$B_{f_1}=[B_{1e} \vdots B_{1t}] \qquad (6-5)$$

其中B_{1e}是非奇异的。再设G_2的一个基本截割组矩阵C_{f2}为

$$C_{f_2}=[C_{2e} \vdots C_{2t}] \qquad (6-6)$$

其中C_{2t}是非奇异的。假使

$$B_{f_1}C_{f_2}^t=0 \qquad (6-7)$$

那么

$$B_{1e}C_{2e}^t+B_{1t}C_{2t}^t=0 \qquad (6-8)$$

因为C_{2t}是非奇异的，它是有逆的，

$$B_{1t}=B_{1e}C_{2e}^t(C_{2t}^t)^{-1} \qquad (6-9)$$

所以

$$B_{f_1}=[B_{1e} \vdots B_{1e}C_{2e}^t(C_{2t}^t)^{-1}]=B_{1e}[U \vdots C_{2e}^t(C_{2t}^t)^{-1}] \qquad (6-10)$$

另一方面，用G_2的一个基本回路矩阵$B_{f_2}=[U \vdots B_{2t}]$，根据式（5－59）

$$B_{f_2}C_{f_2}^t=0 \qquad (6-11)$$

或

$$C_{2e}^t+B_{2t}C_{2t}^t=0 \qquad (6-12)$$

所以

$$B_{2t}=C_{2e}^t(C_{2t}^t)^{-1} \qquad (6-13)$$

或

$$B_{f_2}=[U \vdots C_{2e}^t(C_{2t}^t)^{-1}] \qquad (6-14)$$

因为B_{f_2}是G_2的一个基本回路矩阵，对式（6－14）先乘以任何一个元素值为1或0的非奇异矩阵时，所得的结果矩阵也是G_2的一个基本回路矩阵。因为B_{1e}是一个非奇异矩阵，我们

可以用它来作成G_2的另一个基本回路矩阵

$$B_{1c}B_{f2}=B_{1c}[U \vdots C_{2c}^t(C_{2f}^t)^{-1}] \qquad (6-15)$$

这个矩阵正是式（6—10）所给出的G_1的基本回路矩阵。根据定理6—3，我们证明了G_1和G_2是2-同构的。

定理6—5：设B_{f1}是一个图G_1的一个基本回路矩阵，C'_{f2}是另一个图G_2的一个基本截割组矩阵。当且仅当置换C_{f2}的各列得到C_{f2}而使

$$C_{f2}B_{f1}^t=0$$

时，G_1和G_2是2-同构的。

例6—1　图6—5（a）中G_1的一个基本回路矩阵B_{f1}为

$$B_{f1}=\begin{array}{c} \begin{matrix} a_1 & b_1 & c_1 & d_1 & e_1 & f_1 & g_1 & h_1 \end{matrix} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

图（b）中G_2的一个基本割截组矩阵C'_{f2}为

$$C'_{f2}=\begin{array}{c} \begin{matrix} a_2 & b_2 & c_2 & d_2 & e_2 & f_2 & g_2 & h_2 \end{matrix} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array}$$

将C'_{f2}中的各列置换后得

$$C_{f2}=\begin{array}{c} \begin{matrix} e_2 & f_2 & d_2 & g_2 & h_2 & a_2 & b_2 & c_2 \end{matrix} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

所以

$$C_{f2}\cdot B_{f1}^t=0$$

因此图6—5中的G_1和G_2是2-同构的。

从定义来看，我们知道同构图总是1-同构的，而1-同构图总是2-同构的，但是2-同构图不一定是1-同构，正象1-同构图也不一定是同构的。然而对连通度为3或更多的图来说，同构，1-同构，和2-同构这3个词便是同义词了。

第三节　平　面　图

如果一个图可以画在一个平面上而其中没有两条边是交叉的，这个图就叫做平面图。要是一个图画在一个平面上而有相交叉的边的，就叫作非平面图。

在任何一个平面或曲面上，一个图的几何图形可以画在这个面上而没有出现边相交叉的，就叫作埋置。所以，要是说一个图G是非平面的，我们必须证明G的所有可能的几何表

示形式中没有一个是可以埋置在一个平面上的。同样意义，如果一个几何图G有一个同构图可以埋置在一个平面上，那么这个图G就是平面图，否则，G是非平面图。把一个平面图埋置在一个平面上的方法叫作G的平面表示法。

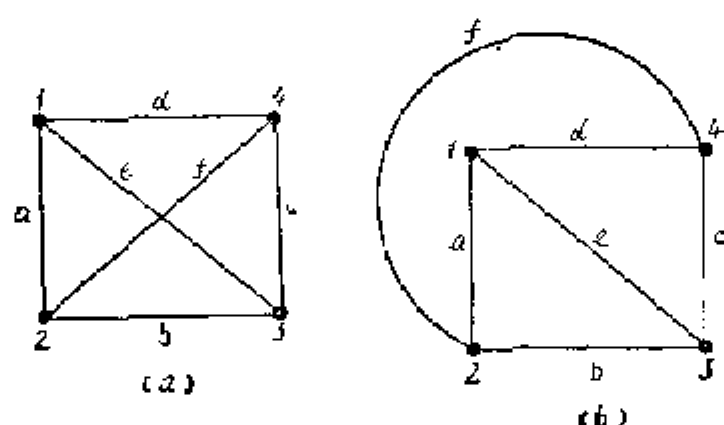


图 6－7

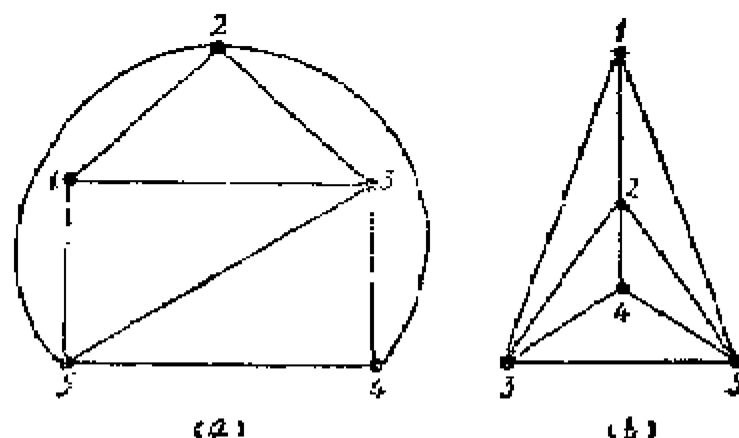


图 6－8

例如图6－7（a），这个图骤看起来不是一个平面图，因为边e和边f是交叉的，所以这个图不能埋置在一个平面上。如果我们把边f（或边e）移画在四边形之外面其它各边仍不变，如图（b）所示，我们就能够把这个新的几何图形埋置在一个平面上了。所以如图6－7（a）所示的图是一个平面图。又象第二章中图2－12所示的两个同构图不过是同一图的不同几何表示法而已，图（a）是平面表示，而图（b）则不是。这个图当然是平面图。另一方面，图2－13中的3个图无论如何不能把它们重画在一个平面上面没有相交叉的边，所以它们是非平面图。

从图6－7（b）看来，似乎在一个平面上要画成一个平面图，问题在于能否用弯曲的线条画出迂回的路线，不使出现边相交叉。事实不是这样的。有一个定理告诉我们，没有必要把一个平面图中的边画成曲线，以避免边相交叉。

定理6－6：任何一个简单平面图能够埋置在一个平面上面每条边可以画成直线段。

这个定理的证明是比较复杂的，这里从略。作为一个例子，可看图6－8。在图6－8（a）中是有弯曲边的，而用直线段重画成图（b）。这个定理中，规定所埋置的图必须是一个简单图，因为自环或两条平行边中的一条就不能用直线段画出来的。

一个图G的平面表示法中，各边把这个平面分成若干区，或称面，如图6－9所示。

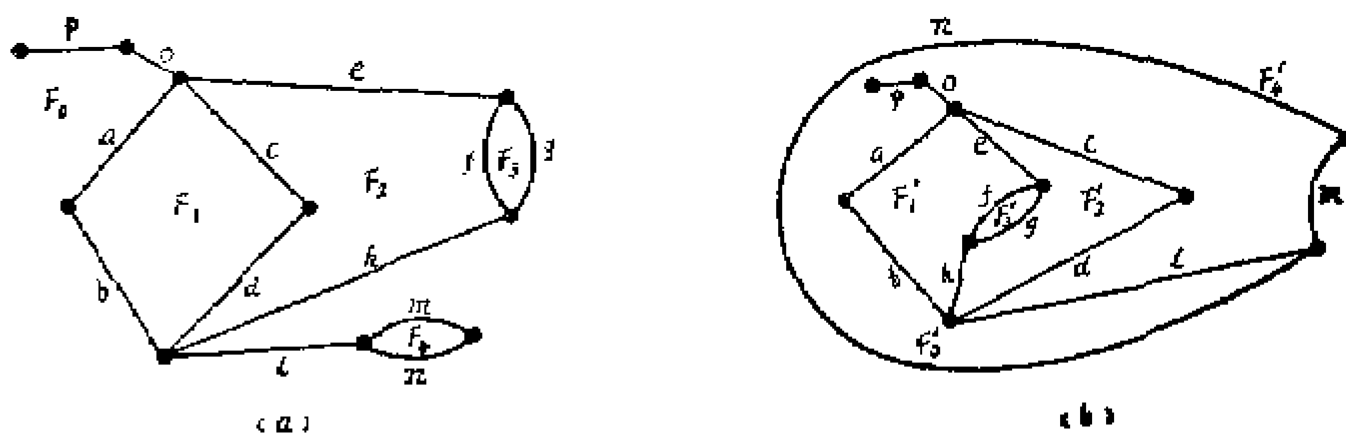


图 6－9

在图（a）中，图G把一个平面分成5个区，即F_0，F_1，F_2，F_3和F_4。一个区由边组（或点组）作为界的。F_0叫作无限区或外区，它是位于埋置在平面上的图以外的区域，广延到无限的。其它区如F_1，F_2，F_3，和F_4叫作有限区或内区。一个图的区组不是唯一不

能变的。如果我们把图6—9（a）重画或另外埋置，如图（b）所示，它的界就有F'_0，F'_1，F'_2，F'_3，和F'_4，是与图（a）中的界不同，原来的有限区F_4变成现在的无限区F'_4。所以改变一个平面图的埋置法，我们可以改变无限区。事实上，用适当的埋置法，我们可以使任何一个区变成无限区。

放置一个球面S在一个平面Z上，如图6—10所示。球面S接触平面Z的那点叫它为南极SP。由SP画一条直线垂直于平面Z，这条直线向上与球面S接触的那点叫它为北极NP。

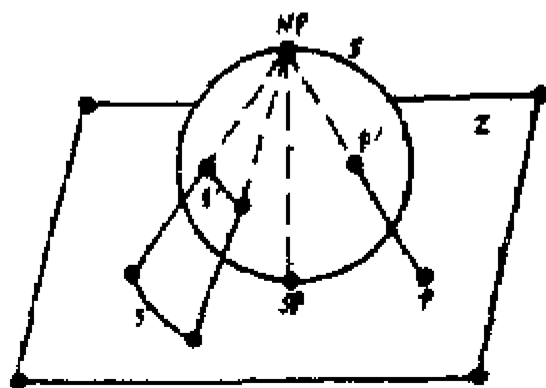


图 6—10

应用球极平面投影法，我们可以把埋置在球面S上的图投射到平面Z上，反之，一个埋置在平面Z上的图也可以投射到球面S上。由平面Z上的点p画一条直线与NP相连，这条直线与球面相交于点p'，这点p'是唯一的，叫作p在球面上的对应点。平面Z的一条线段q也可以在球面上得到一条对应段q'。因此，在球面和平面之间有一一对应的点和边。平面Z上在无穷远的点对应着球面S上的北极NP。由此得一定理。

定理6—7：当且仅当一个图能够埋置在一个平面上时，它就可以埋置在球面上。

当一个平面图埋置在球面上时，将球面分成若干区。在球面上每个区都是有限的，因为平面上的无限区映射到球面上包括北极点的那个区。所以，当我们要使平面上一个平面图的某一区F成为无限区的话，我们可以用上述的投影方法把这个图先映射到球面上。靠适当地旋转球面，使北极点NP正在F_0区里，然后再把球面上的图又画回到平面Z上，那时那个区F_0就变成无限区了。由此我们又得到一个定理。

定理6—8：一个平面图可以埋置在一个平面上，使某一特定区（就是由组成这个区的各边所特定的）成为无限区。

用球面上区的概念来思考，我们知道在平面上的无限区与有限区之间是没有什么真正区别的。所以，当我们讲到一个图的平面表示法时，我们的意思是包括无限区的。同样，既然把一个平面图不论埋置在一个平面上或者一个球面上，其间没有什么本质上的区别（一个平面可以看作是无限大半径的球面）。所谓一个图的平面表示法也常常用来包括平面埋置以及球面埋置。

既然一个平面图可能有不同的平面表示法，我们也许要问每一种埋置法是否有同样的区数。回答是"是的"。下一个定理通称欧拉多面体公式就告诉我们任何一个平面图的区数。

定理6—9：有n_v个点和n_e条边的一个连通平面图有n_e-n_v+2个区。

证：对这个定理，我们只要证明对于简单图是这样就够了。面为增加一个自环或者一条平行边仅仅是一方面增加一个区到这个图，另一方面也同时给n_e数增加了1。我们也可以不管或删除那些并不形成区界的那些边，如图6—9中的边o和p。增加（或删除）任何这样的一条边，一方面使n_e数增加（或减少）1，但另一方面也使n_v数增加（或减少）1，所以仍保持n_e-n_v的数量不变。

因为任何一个简单平面图的平面表示法可以使每条边是一直线（定理6—6），任何一个平面图可以画成每区是个多角形（一个多角网络）。设代表给定图的多角网络包括f个区，并设k_p为p-边区的区数。因为每一条边正好是两个区的界，所以

$$3k_3+4k_4+5k_5+\cdots+rk_r=2n_e \qquad (6-16)$$

其中k_r是边数最多的多边形数。

同样

$$k_3+k_4+k_5+\cdots+k_r=f \qquad (6-17)$$

在多角网络中对向各点的所有角之和为

$$2\pi n_v \qquad (6-18)$$

想到一个p-边多角形所有内角之和为π（$p-2$）和所有外角之和为π（$p+2$）。让我们计算式（6—18）作为（$f-1$）个有限区的所有内角之和再加上规定无限区那个多角形的所有外角之和的总计。这个总计为

$$\begin{aligned}&\pi(3-2)k_3+\pi(4-2)k_4+\cdots+\pi(r-2)(k_r-1)+\pi(r+2)\\&=\pi(3-2)k_3+\pi(4-2)k_4+\cdots+\pi(r-2)k_r+4\pi=\pi(2n_e-2f)+4\pi\end{aligned} \qquad (6-19)$$

使式（6—19）等于式（6—18），得

$$2\pi(n_e-f)+4\pi=2\pi n_v$$

或

$$n_e-f+2=n_v \qquad (6-20)$$

所以区数为

$$f=n_e-n_v+2 \qquad (6-21)$$

由此得一推论：任何一个有f个区，n_v个点，和n_e条边（$n_e>2$）的简单的连通平面图中，下列不等式一定是成立的：

$$n_e\geqslant\frac{3}{2}f \qquad (6-22)$$

$$n_e\leqslant 3n_v-6 \qquad (6-23)$$

证：因为每个区至少有3条边为界，并且每条边正好属于两个区，所以

$$2n_e\geqslant 3f$$

或

$$n_e\geqslant\frac{3}{2}f$$

把欧拉多面体公式代去不等式（6—22）中的f，得

$$n_e\geqslant\frac{3}{2}(n_e-n_v+2)$$

或

$$n_e\leqslant 3n_v-6。$$

不等式（6—23）常常用来检查一个图是否平面的。如果一个图违反了这个不等式，那么那个图是非平面的。可是要注意不等式（6—23）只是一个图的平面性的一个必要条件，而非充分条件。换句话说，虽然每一简单平面图必须满足式（6—23），可是只满足这个不等式不能保证一个图的平面性。我们将在下一节中再谈这个问题。

第四节 两个典型的非平面图

设边e_1的端点为u_1和u_2，边e_2的端点为w_1和w_2，其中$u_1\neq u_2$，$w_1\neq w_2$。如果$u_2=w_1$，并且没有别的边以u_2作为它们的端点，那么边e_1和e_2叫作串联。所谓串联边就是表明这些

边是串联的。例如，在图6—11中，边e_1和e_2是串联的，我们就可以叫这两条边是串联边。

如果在两个图中的一个图，把某些边换以串联边（即在边中插入一些次数为2的点），或者合并一些串联边为一条边，能使这两个图成为同构的，那么这两个图称为同胚的。例如，图6—12中的G_1和G_2就是同胚图。

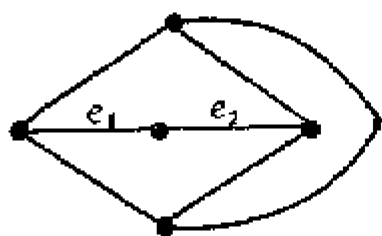
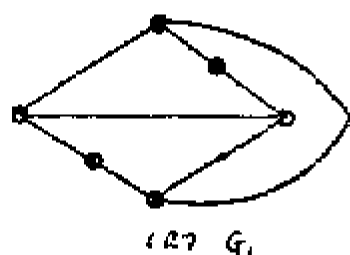


图 6—11



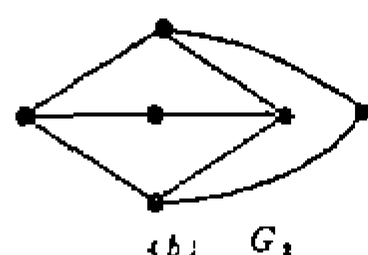


图 6—12

现在我们要学习关于区别一个平面图和一个非平面图的问题。先学习一个定理。

定理6—10：一个有5个点的完全图是一个非平面图。

证，设这个完全图的5个点命名为1，2，3，4，5。我们在上面讲过，所谓一个完全图是一个简单图，其中每个点都有一条边与其它点直接连接的。

图6—13（*a*）是一个5边形，由点1到点2到点3到点4到点5再回到点1成为一个回路。这个5边形把纸的平面分成两个区，一个是5边形内的有限区，一个是5边形外的无限区。

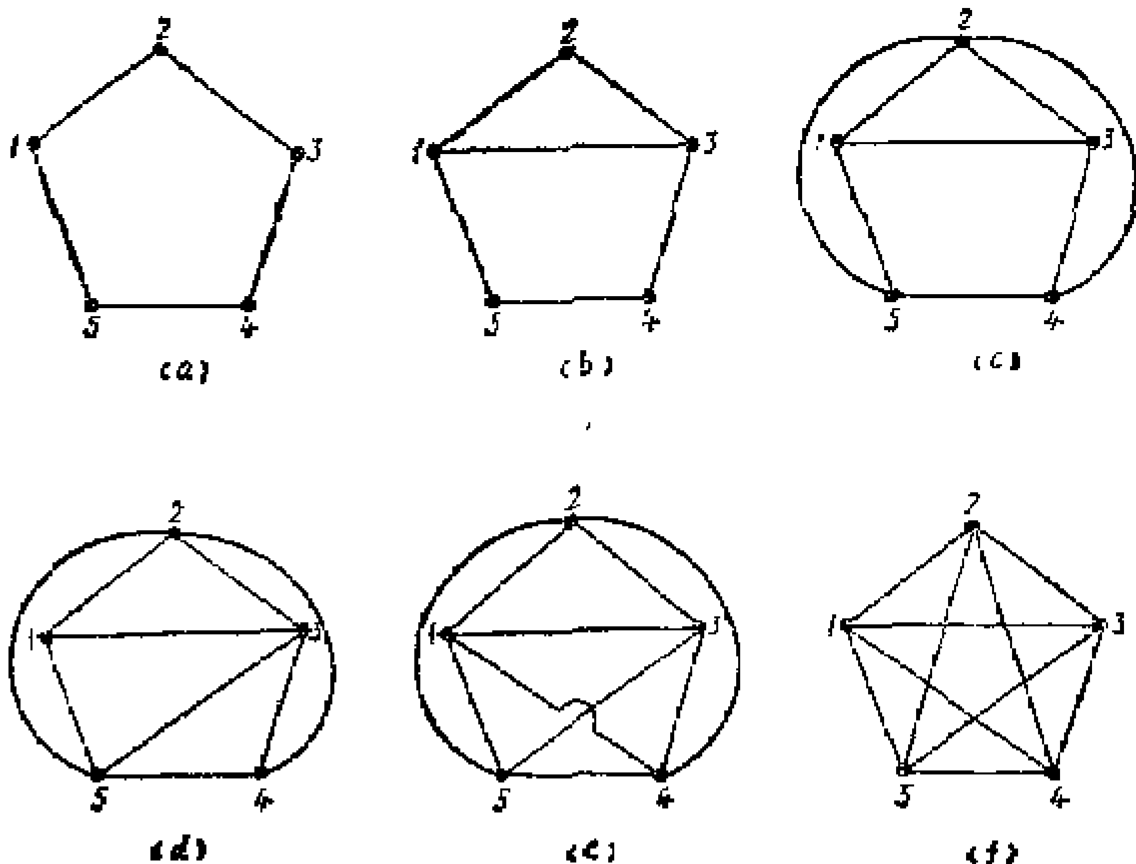


图 6—13

现在我们在点1与点3之间连接一条边，这边可以画在5边形里，也可以画在5边形外（画在外面时，当然不与（*a*）中已经画好的5边相交叉）。假定我们选择画在5边形内，如图（*b*）所示。如果我们选择画在外面，讨论的结果还是一样的。现在我们要再画一条由点2到点4的边，和由点2到点5的边。因为这两条边如果画在5边形里，那么就会与原来已经画好的边相交叉，所以我们把这条边画在5边形外，如图（*c*）所示。现在再画连接点3到点5的边，如果这条边再画在5边形外，就要与点2到点4之间的边相交叉，所以点3与点5之间的边只能画在5边形里，如图（*d*）所示。现在在这个完全图中还有一条连接点

1 到点 4 之间的边要画。这条边不论画在 5 边形里或外，都要与别的边相交叉，如图（e）〔图（f）就是图（e）的另一种画法〕所示。所以说这个图不能埋置在一个平面上的。

这个证明完全是直觉观察的，上一节中我们已经看到了代数证明。在图 6 —13（c）中，这个完全图有 5 个点，即$n_v = 5$和10条边，即$n_e = 10$。$3n_v - 6 = 15 - 6 = 9 < n_e$。它不能满足不等式（6 —23），所以是一个非平面图。

上面这个 5 点的完全图是两个典型的非平面图之一。另一个典型的非平面图是一个3-连通的3-正则图，它的两个通常的几何表示图形如图 6 —14所示，它们是同构的。

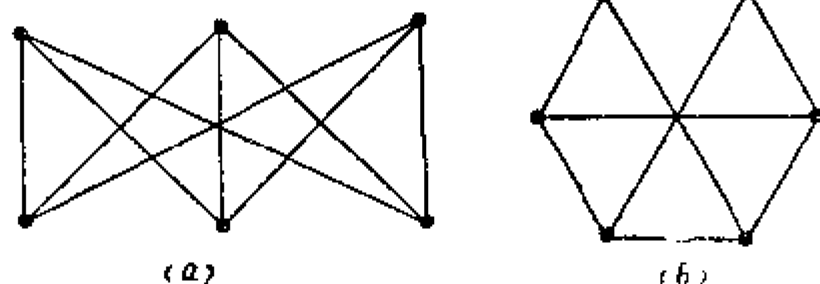


图　6 —14

定理 6 —11：一个 6 个点3-连通的3-正则图是非平画的。

这个定理也可以用如图 6 —13所用的办法来证明它是非平画的。

这两个典型的非平面图叫作柯拉托夫斯基图，它们是由一位波兰数学家柯拉托夫斯基首先发现它们的特性的。在文献中，图 6 —13（f）常标以K_5，图 6 —14标以$K_{3,3}$，K字就是，发现者名字的首字母。

这两个典型的非平面图有下列共性：

（一）都是正则图。

（二）都不能埋置在一个平面上。

（三）删去一条边或一个点，每个都成为平面图。

（四）图 6 —13（f）是点数最少的非平面图，图 6 —14是边数最少的非平面图，两者都是最简单的非平画图。

象图 6 —14（a）那样的图称为两分图。它的点组V可以划分为两组，譬如说，上面的 3 个点称为点组V_1，下面的 3 个点称为点组V_2。由图可见，图中每条边的一个端点与V_1相关联，而另一个端点与V_2相关联。

上面讲图的平面性时，我们都是讲连通图的。对于分离图呢，那就要把各个部分单独考虑。所以一个分离图是否是平面的，就要看它的每个部分是否是平面的。同样，对于可分图（或1-连通图）也像分离图一样，各个部分分开考虑。所以当且仅当一个分离图和一个可分图中每个部分都是平面的，它们才是平面的。

在埋置或平面性问题上，只需考虑不可分图。

在一个球画上埋置图时，是否只有一个埋置法呢？如果把一个平面图用两种办法埋置在一个球面上时，当把一个球面对另一个球面作适当旋转时能使两者叠合的，也可能使某些区发生些畸变，但不使一个点越过一条边时，这两种埋置法是没有差别的。凡是一个图在一个球面上所有可能的埋置法中，没有两种不同方法时，这个图就叫作在一个球面上有一个唯一埋置法，或叫作有一个唯一平面表示法。

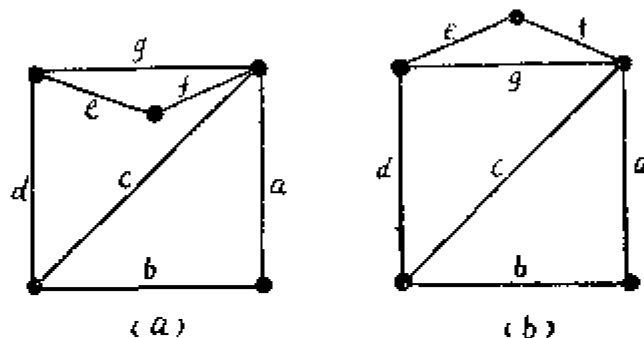


图　6 —15

例如，在图 6 —15中，同一个图有两种埋置法。埋置法（b）有一个区是由 5 条边为界的，即（b）中的无限区是由边组（a，b，d，e，f）为界，但是埋置法（a）中却没

有5条边为界的区。所以旋转分别埋置（a）和（b）的两个球面是无法使它们叠合的。因此这两种埋置法是有差别的，而这个图是没有唯一平面表示法的。但是，如果把图6—8中的两个图分别埋置在两个球面上，适当地旋转球面是可以使它们叠合的。这里要注意的是边是可以弯曲的，和在球面埋置时是没有无限区的。关于唯一埋置法的问题有一个定理如下。

定理6—12：每一平面的3-连通图的球面埋置法是唯一的。

这个定理在确定一个图的平面性时是很重要的。这个定理指出，如果一个3-连通图可以埋置的话，那就只有一种埋置法。

第五节　平面性的检查

如何确定一个给定图G是平面的还是非平面的是一个重要问题。只靠画出图来找寻平面性显然不是一个好办法。我们一定要有些简单而有效的判别准则。为了达到这个目的，我们采取下列的初等简化步骤：

第一步，既然当且仅当一个分离图的每个部分和一个可分图的每个片都是平面的，它们才是平面图，所以我们一次只能考虑一个部分或一片。因此对于某一个给定的任意图G，先要确定它的组成部分

$$G=\{G_1,\ G_2,\ \cdots,\ G_k\}$$

其中每一个G_i是G的一个不可分的片，然后测试每个G_i的平面性。

第二步，既然增加或删除自环不影响平面性，先删除所有的自环。

第三步，既然平行边也不影响平面性，在每一对点之间，除留一条边之外，删除所有其它的平行边。

第四步，靠合并两条串联边来删除次数为二的点不影响平面性，所以删除所有串联边。

反复地进行上述的第三步和第四步一般将大大简化一个图。例如，图6—16说明图6—15（b）的串并联简化法，其中图（a）采用第四步把串联边简化掉，图（b）采用第三步把平行边简化掉，图（c）采用第四步把串联边简化掉，图（d）采用第三步把平行边简化掉。

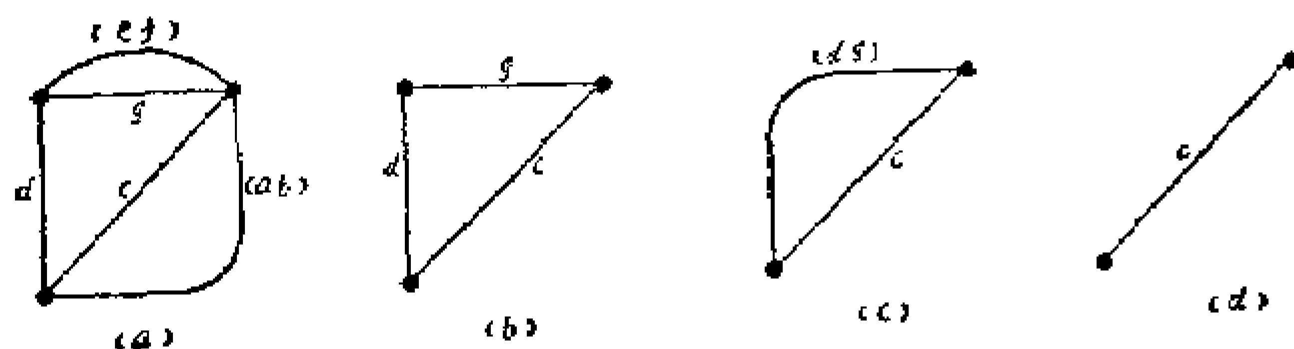


图　6—16

设一个不可分图G_i通过反复采用第三步和第四步后简化到一个新图H_i，新图H_i会成为怎样的呢？下一定理将回答这个问题。

定理6—13：由一个不可分图通过上述简化步骤所得的新图H_i有3种可能，就是：

（一）一条单边；

（二）一个有4个点的完全图；

（三）一个不可分的简单图，它的$n_v\geq5$，$n_e\geq7$。

在定理6—13中，所有属于第1类或第2类的新图H_i是平面的，用不着再行检查。

所以，今后我们需要研究的只是那些至少有5个点和每点的次数为3或更多的简单的，连接的，不可分图。下一步，我们可以根据式（6—23），即$n_e \leqslant 3n_v - 6$来检查。如果不能满足这个不等式，那么图是非平面的。如果能满足这个不等式呢？我们在上一节中已经知道，这个不等式只是一个必要条件，而非充分条件，所以还需作进一步检查。例如，非平面图$K_{3,3}$能满足这个不等式，因为$K_{3,3}$有9条边，6个点，即$n_e = 9$，$n_v = 6$，所以$3n_v - 6 = 18 - 6 = 12 > n_e$，这是满足这个不等式的，可是$K_{3,3}$是个非平面图。

为了证明$K_{3,3}$的非平面性，我们可以利用另一事实，即这个图没有一个区是由少于4条边为界的。所以，要是这个图是平面的，我们会有

$$2n_e \geqslant 4f$$

由欧拉多面体公式代去上式的f，得

$$2n_e \geqslant 4(n_e - n_v + 2)$$

或

$$2 \times 9 \geqslant 4(9 - 6 + 2)$$

或

$$18 \geqslant 20$$

这是一个矛盾。所以这个图不可能是平面的。

为了更确实地进一步检查，那就要用到下一个很重要的定理了。

定理6—14：对于一个图G是平面图的一个必要和充分的条件是：G不含有象典型的非平面图中任何一个图，或与这两个非平面图同胚的图。

这里有一点值得注意，就是对于一个非平面图，不是必需有象典型的两个非平面图中任何一个作为它的子图，但是可能有一个子图是一个典型非平面图的同胚。例如，图6—17（a）有一个子图如图（b）所示，后者是图6—14的同胚。所以图6—17（a）是非平面的。又例如，图6—18（a）是非平面的，可是没有一个子图是属于典型非平面图的。不过，如果我们从图（a）中删除边a和边b，则得如图（b）那样一个子图。把图（b）中的串联边c和d在点3合并为一，则得图（c）。图（b）是图（c）的同胚。很明显，图（c）是图6—14的同构，这就证明图6—18（a）是非平面的。

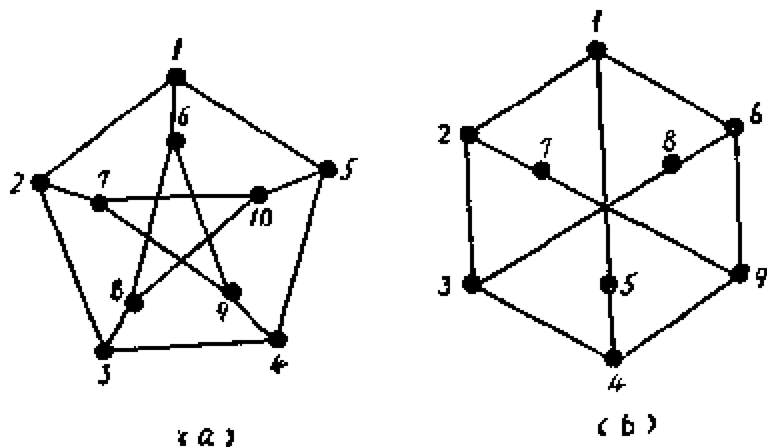


图 6—17

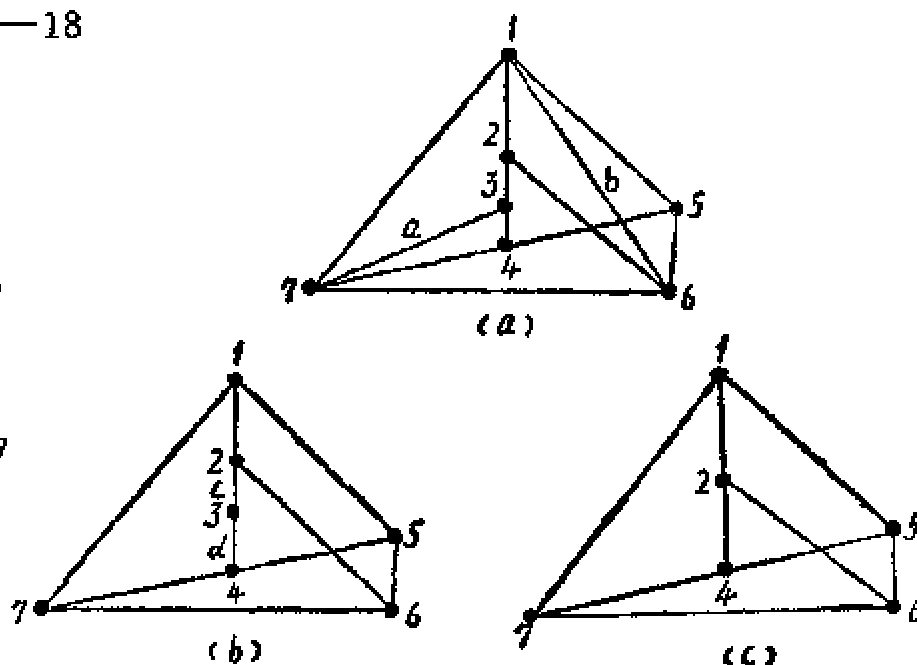


图 6—18

第六节 对 偶 图

虽然定理6—14对一个图的平面性是一个极好的和看起来简单的判别准则，可是应用它

来测试一个大图，譬如说，一个有25个点的简单不可分图，而每个点的次数为3或更多些，却是困难的。对一个平面图，另有几种可供采用的表征，其中之一就是是否存在对偶图的问题。

在第三章中，由式（3—11）我们知道一个图G的圈数N是

$$N = n_e - n_v + k = n_e - R \qquad (6-24)$$

所以

$$N + R = n_e \qquad (6-25)$$

例如，在图6—17（a）中的连通图G有10个点和15条边。所以G的圈数N是6。从式（5—46）知一个连通图的基本回路矩阵的秩数R为n_e-n_v+1，也是6。所以一个连通图的圈数等于它的基本回路矩阵的秩数。

本章在讨论对偶图中，我们将包括那些有自环作为边的图在内。

设一个图G_1包括边a，b，…，n_e，另一个图G_2包括a'，b'，…，n'_e。假定我们选择G_2中的边p'作为那个对应于G_1中边p的边，其中$p=a$，b，…，n_e。这样，我们在G_1的边与G_2的边之间就有一个一一对应关系。设g_1是G_1的一个子图，包括边i_1，i_2，…，i_k。那么包括边i'_1，i'_2，…，i'_k的G_2的那个子图g_2就叫作g_1的对应子图。同样，为了方便，用符号$\overline{g}$来表示一个图G的子图g的余图，这就是，$\overline{g}$包括G中除g的所有边以外的其它边，以及不在g中的孤立点。

例6—2　图6—19中（a）和（b）分别是两个图G_1和G_2。假定G_1中的边p对应着G_2中的边p'，其中$p=a$，b，…，g。再假定我们选择G_1的一个子图g_1，它包括边a，b，…，e。如图6—20（a）所示，那么G_2中对应着g_1的子图g_2，就是包括边a'，b'，…，e'的子图g_2，如图6—20（b）所示。另外g_1的余图$\overline{g_1}$将包括边d，f和g，如图6—20（c）所示；g_2的余图$\overline{g_2}$将包括边d'，f'和g'，如图6—20（d）所示。有了这些子图，我们可以对图的对偶性下定义了。

图 6—19

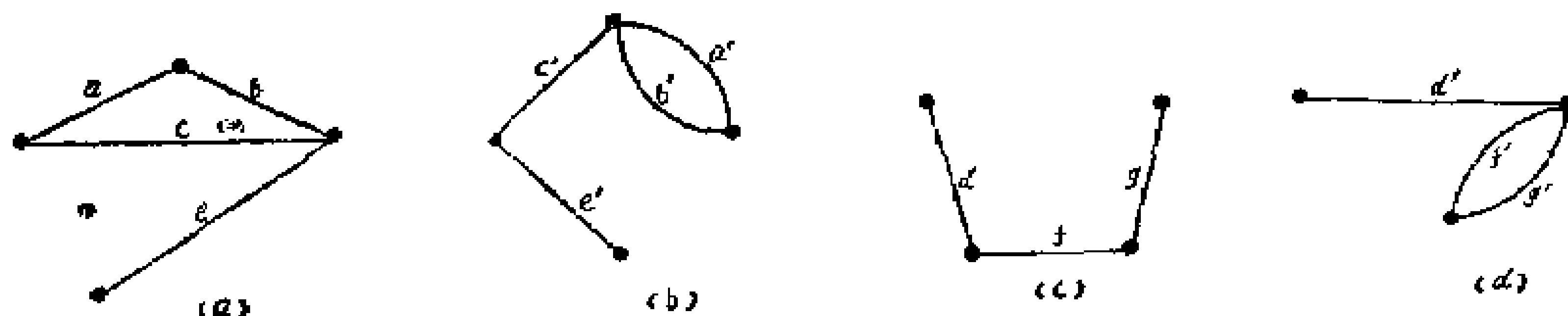


图 6—20

如果有两个图G_1和G_2，在G_1和G_2的各边之间有着一一对应关系，使得G_1的每一个子图g_1的

$$n_1 = R_2 - \overline{r_2} \tag{6—26}$$

其中n_1是g_1的圈数，R_2是G_2的秩数，$\overline{r_2}$是$\overline{g_2}$的秩数，g_2是G_2的子图对应着g_1以及$\overline{g_2}$是g_2的余图，那么G_2就叫作G_1的对偶。

根据这个定义，如果$g_2 = G_2$，那么$\overline{g_2}$就没有边了。因此$\overline{r_2}$是零，并且$g_1 = G_1$，这就意味着$n_1 = N_1$（N_1是G_1的圈数）。这样，如果G_2是G_1的对偶，由式（6—26）得

$$N_1 = R_2 \tag{6—27}$$

因为当G_2是G_1的对偶时，G_1和G_2中的边数n_e是相同的，所以式（6—27）可以写作

$$n_e - N_1 = n_e - R_2 \tag{6—28}$$

根据式（6—27），式（6—29）即成为

$$R_1 = N_2 \tag{6—29}$$

所以，根据式（6—27）和（6—29），如果G_1和G_2是对偶的，那么，$N_1 = R_2$和$R_1 = N_2$。

设g_1是G_1的一个子图，它有n_{e1}条边。设n_{e2}是

$$n_{e2} = n_e - n_{e1} \tag{6—30}$$

或

$$n_{e1} + n_{e2} = n_e \tag{6—31}$$

其中n_e是G_1的边数。当G_2是G_1的对偶时，则G_2中g_2的边数是n_{e2}，其中g_2是$\overline{g_2}$的余图，$\overline{g_2}$是对应着$\overline{g_1}$的。那么式（6—26）成为

$$\overline{n_1} = R_2 - r_2 \tag{6—32}$$

其中$\overline{n_1}$是$\overline{g_1}$的圈数，r_2是g_2的秩数。式（6—32）可以表达为

$$n_e - \overline{n_1} = n_e - R_2 + r_2 \tag{6—33}$$

或根据式（6—19）和（6—31）

$$n_{e2} + n_{e1} - \overline{n_1} = N_2 + r_2 \tag{6—34}$$

将项n_{e2}移至等号的右方，并根据式（6—19）得

$$\overline{r_1} = N_2 - (n_{e2} - r_2) \tag{6—35}$$

或

$$\overline{r_1} = N_2 - n_2 \tag{6—36}$$

根据式（6—29），式（6—36）成为

$$n_2 = R_1 - \overline{r_1} \tag{6—37}$$

这就表示如果G_2是G_1的对偶，则G_1是G_2的对偶，即G_1和G_2是互为对偶的。

对一个平面图G_1，我们得到下述一种方法来构成对偶于G_1的另一个平面图G_2。

设有n_e条边的G_1是一个平面图，它能够埋置在一个平面上而没有边是互相交叉的，在G_1的各区中放上一个点p，然后按照下列方法把各点连接起来。

如果两个区F_i和F_j是邻接的，就是有一条共同边的，画一条虚线连接p_i和p_j，它与F_i和F_j间的共同边只能交叉一次。如果在F_i和F_j之间有一条以上共同边的，则在每一条共同边的p_i和p_j之间画一条虚线连接线。对于完全在一个区F_k的那条边e，则在点p_k上画一个虚线自环，与边e只交叉一次。

用这种方法，我们得到一个新图G_2。这个图G_2就是图G_1的对偶图。G_1和G_2之间的各边是一一对应的。G_1的一边与G_2的一边相交叉。严格地讲，这样构成的对偶是几何对偶。这种找对偶的方法叫作D-方法。

图6—21（a）是一个连通图G_1，它有6个区F_1，F_2，F_3，F_4，F_5，和F_6。各区之中分别放有点1，2，3，4，5和6。图（b）中的虚线图就是通过D-方法所得到的对偶图G_2。

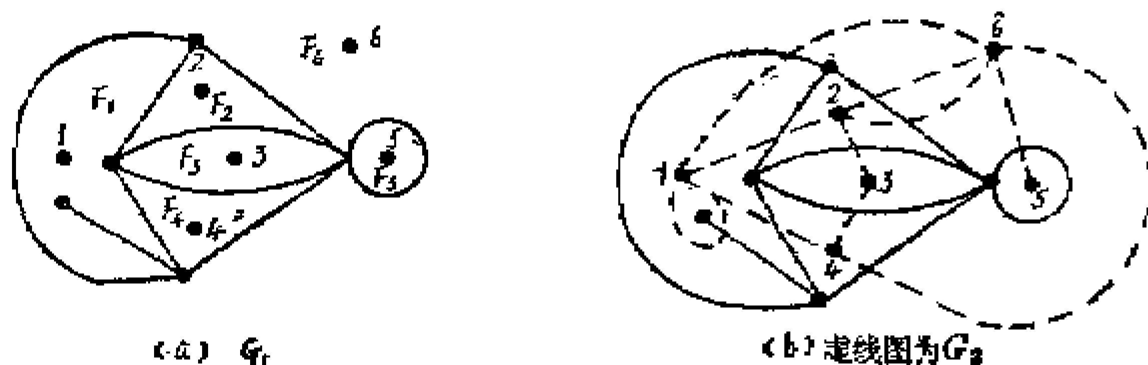


图 6—21

观察图6—21中的两个图，我们得到一个平面图G_1和它的对偶G_2之间的关系如下：

（一）在G_1中形成一个自环的一条边在G_2中产生一条悬挂边。

（二）在G_1中的一条悬挂边在G_2中产生一个自环。

（三）在G_1中的串联边在G_2中产生平行边。

（四）在G_1中的平行边在G_2中产生串联边。

（五）G_1中组成F_i区界的边数等于G_2中对应点p_i的次数，如区内有悬挂边，则沿悬挂边的两面往返，作为两条边计算；反过来也是这样。

（六）G_1可以埋置在一个平面上；G_2也可以埋置在一个平面上，所以也是平面的。

（七）根据D-方法作对偶图的过程，G_1也是G_2的对偶。所以G_1和G_2共称为对偶图。

（八）如果n_{v1}，n_{e1}，f_1，R_1和N_1分别表示一个连通平面图G_1的点数、边数、区数，秩数和圈数，n_{v2}，n_{e2}，f_2，R_2和N_2分别为对偶图G_2中对应的上述各数，那么

$$n_{v2}=f_1 \qquad (6—38)$$

$$n_{e2}=n_{e1} \qquad (6—39)$$

$$f_2=n_{v1} \qquad (6—40)$$

利用上述关系，我们立即可以得到上面已经得到的$R_2=N_1$和$N_2=R_1$。

图6—22是另一个应用D-方法由G_1得到对偶G_2的例子。图（a）是一个平面图G_1，而图（b）中的虚线所示的图是G_1的对偶图G_2。

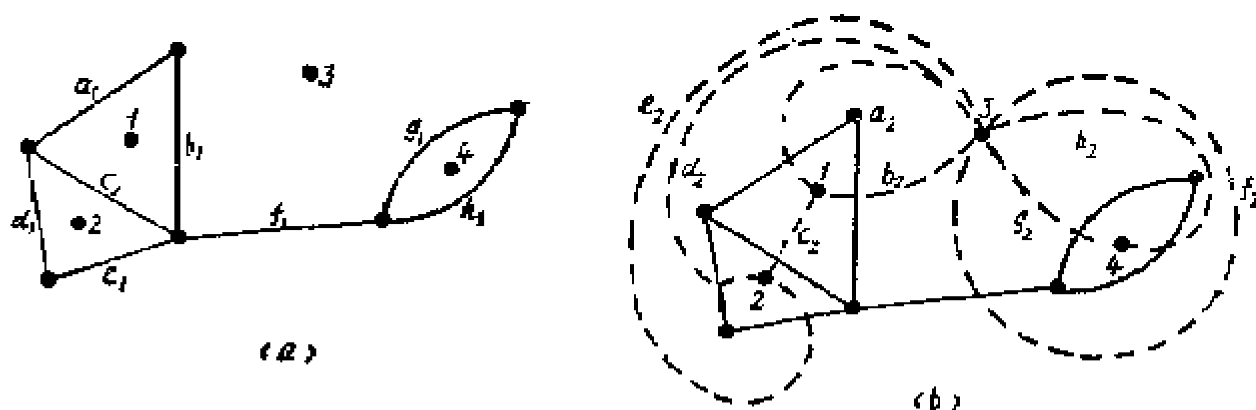


图 6—22

由D-方法所得到的对偶有一个定理：

定理6—15：由一个图G_1通过D-方法而得到的一个图G_2是连通的。

用D-方法是否能够得到任何一个对偶图呢？上面这个定理说，由D-方法所得到的对偶图是连通的。那么，如果一个图（本身是分离的）有一个对偶图，后者就不能由D-方法获得。我们知道这种对偶图是存在的。例如，如果我们选择一个分离图G_1而用D-方法来求它的对偶图G_2，那么G_1和G_2就是互为对偶的。这就是说，G_1是G_2的对偶，而G_1是分离的。所以用D-方法是不能够得到任何一个对偶图的。

通过D-方法由G_1而得到它的对偶G_2时，一个图中的回路与另一个图中的截割组之间有下列关系：

定理 6—16：设G_1和G_2为对偶图。设e_1是G_1中的一条边，e_2是G_2中的对应边。再设g_1是从G_1中短接去e_1所得到的图，g_2是从G_2中删除e_2所得到的图。那么g_1和g_2也是对偶图。

现在讨论一下对偶图的唯一性。从一个图用D-方法所得到的对偶是否是唯一的？换句话说，一个图的所有对偶是否是同构的？从D-方法来看，当且仅当一个图有一个唯一平面表示法或在一个球面上的唯一埋置法时，我们有理由希望一个平面图将有一个唯一的对偶。

例如，在图 6—15中，同一个图（同构的）有两种不同的埋置法（a）和（b）。所以这些同构图的对偶是非同构的，如图 6—23所示，其中图（a）是图 6—15（a）的对偶，图（b）是图 6—15（b）的对偶。这两个图虽然不是同构的，却是 2-同构的。从这个例子推广可以得到下一个定理。

定理 6—17：一个平面图G的所有对偶都是 2-同构的，倘然一个图对G的对偶是 2-同构的话，它也是G的对偶。

有了这个条件，一个对偶可以合适地认为是一个平面图的对偶。

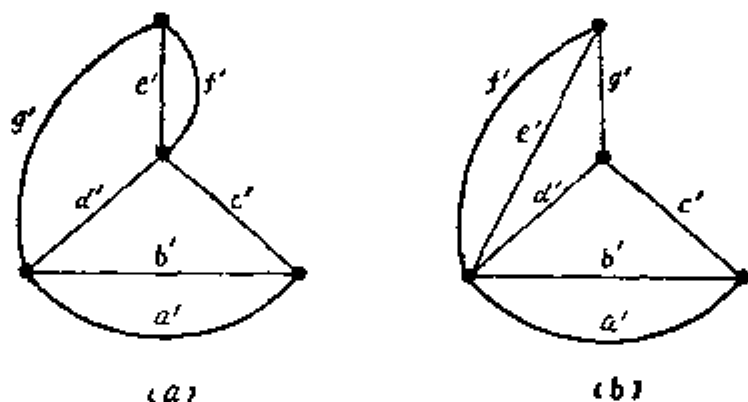


图 6—23

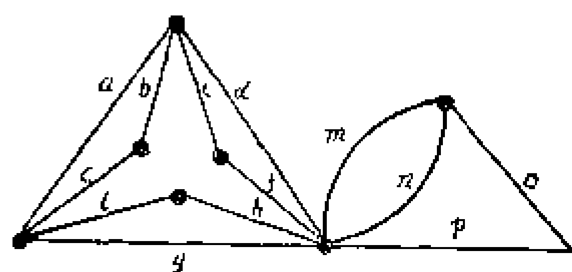


图 6—24

例 6—3　有一个图G如图 6—24所示。通过D-方法，得到G的对偶G_1如图 6—25（a）中虚线所示。将（a）重画得图（b）。图 6—26所示的G_2也是G的一个对偶，可以看出图 6—25（b）的G_1与图 6—26的G_2是 2-同构的。

现在我们要学习关于对偶图的一个重要定理。

定理 6—18：对于两个平面图G_1和G_2互为对偶的一个必要和充分条件是：G_1中的那些边与G_2中的那些边有一一对应的关系，而且当且仅当G_2中的一个边组构成一个截割组时，G_1中的对应边组构成一个回路。

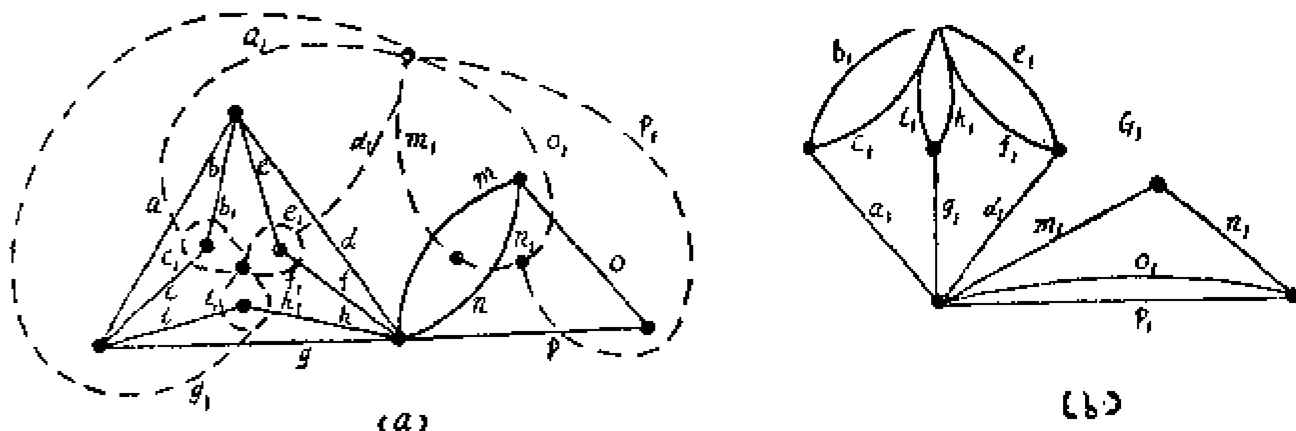


图 6—25

证：让我们考虑一个平面图G的平面表示法，并且也让我们用D-方法画出G的一个对偶G'。然后在G中假定有一个回路C。很明显，C会在G的平面表示中形成某个闭合简单曲线，把这平面分成两个区，一个是图形内的有限区，一个是图形外的无限区。因此G'中的那些点划分为两个非零的，互不相混的子组，一个在C内，一个在C外。换句话说，对应着G中C的那组在C'的那个边组是G'中的一个截割组。同样，对应着G'中截割组S'的，在G中有一个唯一的回路，包括那个对应边组S，使S是一个回路。这就证明了这个定理的必要部分。

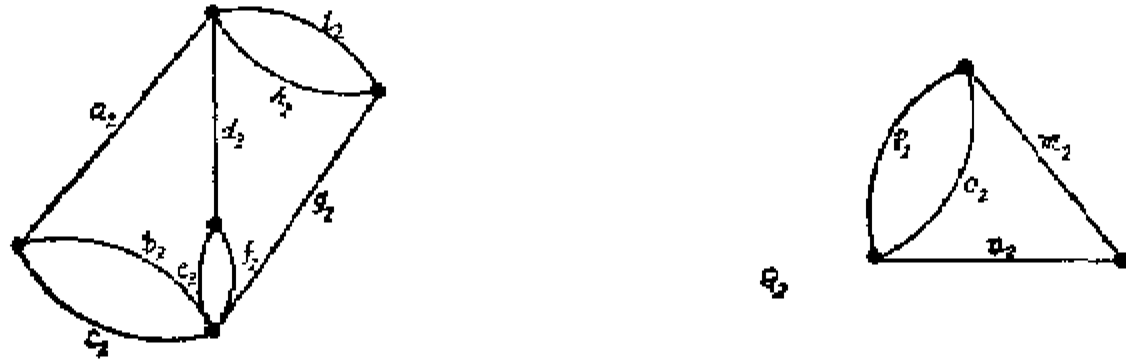


图 6—26

为了证明充分部分，设G是一个平面图，并设G^*是一个图，在G中的那些截割组与G^*中的那些回路之间，或者在G中的那些回路与G^*中的那些截割组之间有着一对一的对应关系。设G'是G的一个对偶图。在G^*中的那些回路与G中的那些截割组之间，以及在G中的那些截割组与G'中的那些回路之间有着一对一的对应关系。所以在G'和G^*的那些回路之间有着一对一的对应关系，这就是意味着G'和G^*是2-同构的（定理6—5）。根据定理6—17，G^*一定是G的一个对偶。

定理6—19： 设G_1和G_2是对偶图，那么G_1中的一个回路对应着G_2中的一个截割组。

现在我们要研究一个子图的对偶。设G_1是一个平面图，而G_2是它的对偶。设a_1是G_1中的一条边，而在G_2中的对应边是a_2。假使我们从G_1中删除边a_1，并且试着找出（$G_1 \ominus a_1$）的对偶，如果边a_1是在两个区的界上，删除边a_1会合并这两个区为一个区。因此（$G_1 \ominus a_1$）的对偶可以从G_2中删除对应边a_2，并且在（$G_2 \ominus a_2$）中合并边a_2的两个端点而得到的。另一方面，如果边a_1不在区界上，那么a_2就是一个自环。这时（$G_2 \ominus a_2$）就是（$G_1 \ominus a_1$）的对偶。因此如果图G_1有一个对偶G_2的话，G_1的任何一个子面的对偶可以连续采用这种步骤来得到的。

至于同胚图的对偶呢？我们设G_1是一个平面图，和G_2是它的对偶。设a_1是G_1中的一条边，而a_2是G_2中的对应边。假使我们在边a_1中增加一个次数为2的点（就是现在一条边a_1变成串联的两条边）。增加一个点将怎样影响对偶图呢？这在G_2中只是增加一条平行于a_2的边。同样，将两条串联边合并为一条边，在G_2中只是减少两条对应的平行边中的一条。因此如果一个图G_1有一个对偶G_2，任何一个同胚于G_1的对偶可以从G_2中用上述步骤得到的。

到此为止我们只研究了平面图的对偶性。这也有道理，因为讲到对偶性或对偶图的唯一性时是建立在平面上的唯一埋置法的。可是，现在定理6—18使我们有对偶性的一个等效的抽象定义，就是，在回路与截割组之间的对应关系，它不是依据一个图的平面表示法。这就使对偶性的概念可以扩展到非平面图。换句话说，给出一个非平面G_1，我们能否找出另一个图G_2，它们两者的各边间有否一一对应关系，使G_1中的每一个回路对应着G_2中的唯一截割

组，或者反是呢？这个问题的答案是"否"。下面就有关于这方面的一个重要定理。

定理6—20：当且仅当一个图是平面图时，它才有一个对偶。

证：要证明这个定理的第一部分，即当它有对偶的问题是容易的，只要用D-方法。

证明这个定理的第二部分，即"仅当"它是一个平面图时，我们只要证明一个非平面图是没有对偶的。设G是一个非平面图，那么根据第四节，G一定含有K_5或$K_{3,3}$，或者一个图同胚于两者之一。根据定理6—14和定理6—18，我们已经知道仅当G的每一个子图g以及与g同胚的每一个图有一个对偶时，图G才能够有一个对偶，因此如果我们能够证明K_5或$K_{3,3}$是没有对偶的，我们就证明了这个定理。

假使典型非平面图$K_{3,3}$有一个对偶D。根据定理6—19，$K_{3,3}$中的一个截割组对应着D中的一个回路，或反是。因为$K_{3,3}$没有只包括两条边的截割组，D也没有只包括两条边的回路。就是说D不含有平行边对。因为$K_{3,3}$中的每条路径的长度有4段或6段，D就没有一个截割组少于4条边的。所以D中每个点的次数至少为4。既然D没有平行边，每个点的次数至少为4，D一定至少有5个点，每个点的次数为4或更多。这就是说D一定至少有（5×4）/2＝10条边。因为我们知道$K_{3,3}$有9条边，那么它的对偶也一定有9条边。上面既然说D至少有10条边，这是矛盾的。所以说$K_{3,3}$不可能有一个对偶。

同样，假使典型非平面图K_5有一个对偶H。注意K_5有（1）10条边；（2）没有平行边对；（3）没有两条边的截割组；（4）只有4条边或6条边的截割组。因此，对偶图H一定有（1）10条边；（2）没有一个点的次数少于3；（3）没有平行边对；（4）只有4段或6段的回路。现在H含有一个6边形（段数为6的一个回路）；参看图6—14（b）对这个6边形，不能增加多于3条边而不产生有3段的回路或是一对平行边。因为这两者对于H都是禁止的，和H有10条边，所以H一定至少有7个点。每个点的次数至少为3。这就导致H至少要有11条边。这是自相矛盾的。

在对偶图中，还有一种自对偶图。如果一个平面图G是它自己对偶的同构，G就叫作一个自对偶图。一个4个点的完全图是一个自对偶图。自对偶图有许多有趣的特性，但也提出一些未解决的问题。

第七节　基础回路全组

如果一个图的一个回路组有下列两个特性：（1）图中每一个回路可以用这个回路组中的几个或所有回路的环和来表达；（2）这个回路组中任何一个回路不能用组中其它一些回路的环和来表达，则这个回路组叫作基础回路全组。基础回路是与第三章，第七节中所讲的基本回路有所区别的。基本回路系对一个图的某一棵生成树而言的，而基础回路不一定是与生成树有关的。不过，这两者却有下述的关系，就是一个基本回路组常常构成一个基础回路全组，而对于所有线性图来讲，一个基础回路全组不一定构成一棵生成树的一个基本回路组。

在一个平面图中，一个基础回路全组另有一种特性：在一个平面的连通图G的平面表示法中，由各有限区回路组构成一个基础回路全组，对于G中的任何一个基础回路全组C可以表达为C中所含规定那些有限区的那些回路的环和，可以观察到图中每一条边最多出现在这种基础回路全组中的两个回路之中，因此，对于每一个平面图G，我们可以找到一个基础回路全组，没有一条边会出现在这些基础回路中的两个回路以上，这个结果使我们又得到一个检

查图的平面性的重要准则。

定理 6 —21：当且仅当一个图G有一个基础回路全组（就是它们所有的N个回路，这个N系G的圈数），没有一条边会出现在这些回路中的两个以上时，这个图G才是平面的。

表示一个平面图的特征有 3 个重要表征，那就是定理 6 —14、定理 6 —18、和定理 6 —21。可是这 3 个表征都有两个缺点：第一，它们应用于一个大图时都很困难；第二，当一个图是平面型时，它们都不会给出图的平面表示。这些缺点近年来鼓励了一些作图法的创制，使得平面性的检测可以根据图的平面表示法。有些作图法是由于应用了数字计算机而完成的。这些方法中的大部分是先把一个给出的图简化到一个或几个简单的不可分图，每一个点的次数为 3 或大于 3，和边数$n_e \leqslant 3n_v - 6$，然后应用作图算法来求得一个图是否是平面的，有些我们将在第十一章第七节中关于平面性测试算法中讲。

第八节 厚度和交叉

如果一个图G是非平面的，就有一些另外的有关表征，象厚度和交叉数等问题。这是因为目前在复杂系统的自动布线，印刷电路工艺，大规模集成电路的设计等领域中，对这些问题是很感兴趣的，而且有一些问题还没有解决。

为了埋置一个非平面图G，就要把G分成若干个平面的子图，分别埋置在几个平面上，而合成一个非平面图G的最少的平面子图数就叫作G的厚度。例如，在一块印刷电路板上，分隔不同子图所需要的绝缘层就是所考虑的非平面图G的厚度。

根据定义，一个平面图的厚度是 1 。典型的两个非平面图的厚度为 2 。试画草图，就可以知道一个有 8 个点的完全图的厚度为 2 ；有 9 个点的完全图的厚度为 3 ，一般地讲，一个任意图的厚度是难于确定的。

习　题

6 — 1　判断图 6 —27中的 3 个图，哪是平面图？哪是非平面图？把平面图画在一个平面上。

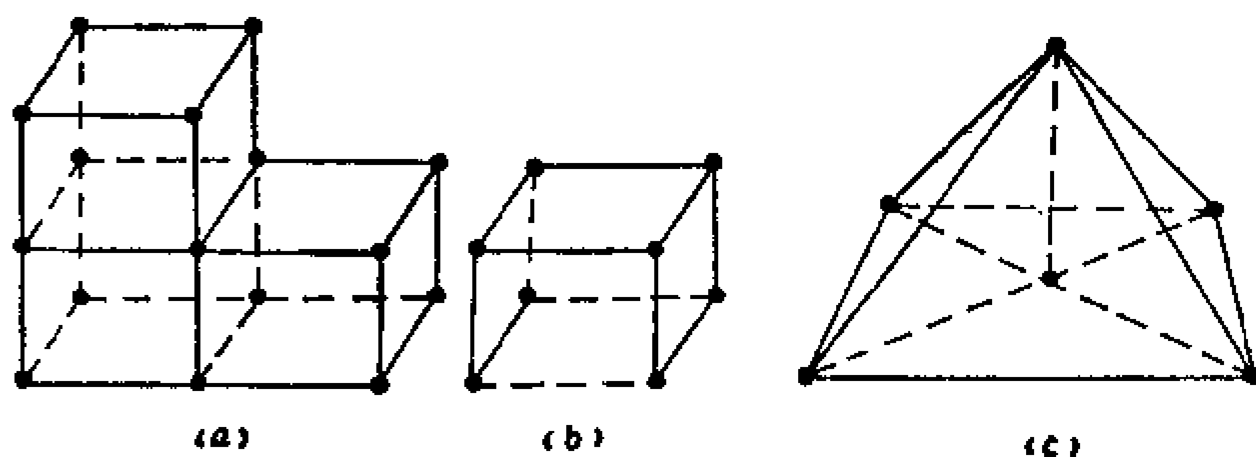


图 6 —27

6 — 2　重画图 6 —28，使图中斜线区成为无限区。

6 — 3　画出图 6 —29的几何对偶图。

6 — 4　利用类似证明定理 5 — 1 的几何理论，证明图$K_{3,3}$也是非平面。

6 — 5　如果有n_v个点，n_e条边的简单平面图，其每个区都是由k条边构成的，试证明

$$n_s = \frac{k(n_v - 2)}{k - 2}$$

6—6　如果平面图G的所有点的次数部相等，构成每个区的边数也相等，则平面图G称为完全正则图。例如图2—28（a）和图2—30（b）是完全正则图。试证明除次数少于或等于2的图外，只存在5种可能的简单的完全正则平面图，并绘出这些图（利用欧拉多面体公式）。

6—7　用画图方法说明：8个点的完全图的厚度为2；9个点的完全图的厚度为3。

6—8　说明4个点的完全图是自对偶的。试再画出两个自对偶图。

6—9　试说明最大平面图的每个区域都是由3条边构成的。

6—10　用定理6—14说明图6—30是非平面图。

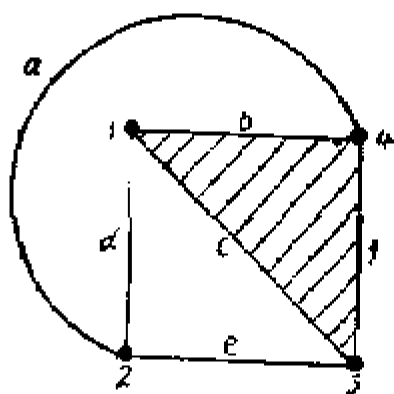


图　6—28

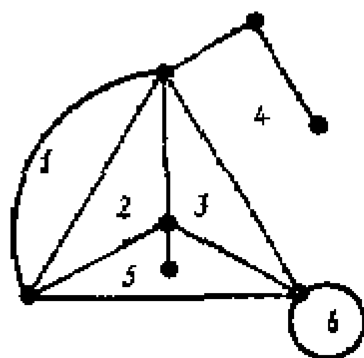


图　6—29

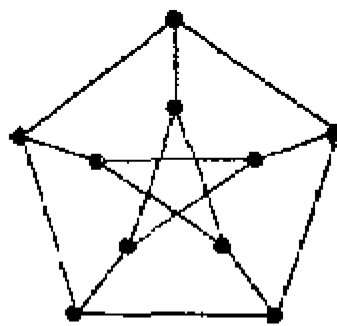


图　6—30

第七章 有 向 图

第一节 有向图的特点和类型

前面几章所研究的线性图都是无向图，每条边并不规定方向。点v_i和v_j两点之间的一条边e_k可以看作是由点v_i走向点v_j，也可以看作是由点v_j走向点v_i的。这一章我们将着重研究每条边有箭头表示的规定方向的线性图，即有向图。这是因为在很多实际情况中都需要有向图。城市的街道常标有单行道；管道流网络中有阀门；电力两络中有输电方向；这些都得用有向图来表示。有向图也是计算机程序所常用的抽象表示，又是研究时序开关网络的有力工具。在控制理论中，也用有向图作为系统分析的信号流图。

在研究无向图中所用的概念和术语，大部分对有向图还是适用的。譬如，一个图的平面性是不管那图是有向的还是无向的，都是一样的。可是，当有向图与无向图有区别时，我们在下定义时，如对行走、路径、回路、树、截割组等就会有所说明；但是无向图中若干特性还是在有向图中有所反映的。

有些情况，在研究有向边时可用无向边来代替，反过来也是这样。所以在研究有向图时，有时我们必须知道所对应的无向图。所谓一个有向图G_0的对应无向图G_n就是把前者每一有向边代以无向边。例如，图7—1（a）是一个有向图，而图（b）就是它的对应无向图。同样，如果一个有向图的对应无向图是连通的，这个有向图就是连通的。例如，因为图7—1（b）是连通的，所以图（a）是连通的。同样，所谓可分有向图，分离有向图，子图和部分等，也是以它们对应的无向图为准的。

图 7—1

在一个有向图中，一条边不仅关联着一个点，而且要看是关联出还是关联入一个点。凡是一条边e_k关联出的那个点v_i叫作边e_K的起点，它关联入的那个点v_j叫作边e_K的终点。像图7—1（a）中，点1是边a的起点，而点3是边a的终点。对于一个自环，像图（a）中的边e，点2既是它的起点，又是它的终点。有的书中把有向图的边叫作弧，以别于无向图中的边。但是在本书中，我们简单地都叫作边，有必要时，就明白地说有向边或无向边。

在有向图中，关联出点v_i的次数叫作点v_i的出次数，用符号$d^+(v_i)$来表示。关联入点v_j的次数叫作点v_j的入次数，用符号$d^-(v_j)$来表示。例如，在图7—1（a）中

$d^+(1)=3 \qquad d^-(1)=1$

$d^+(2)=1 \qquad d^-(2)=2$

$d^+(3)=1 \qquad d^-(3)=4$

$d^+(4)=1 \qquad d^-(4)=3$

$d^+(5)=4 \qquad d^-(5)=0$

所以，在一个有向图G中，入次数的总和等于出次数的总和，而且每个总和等于 G 中的边数，即

$$\sum_{i=1}^{n} d^+(v_i)=\sum_{j=1}^{n} d^-(v_j)=n. \qquad (7-1)$$

一个孤立点的入次数和出次数都等于零。有向图中的一个悬挂点v的次数为1，即

$$d^+(v)+d^-(v)=1 \qquad (7-2)$$

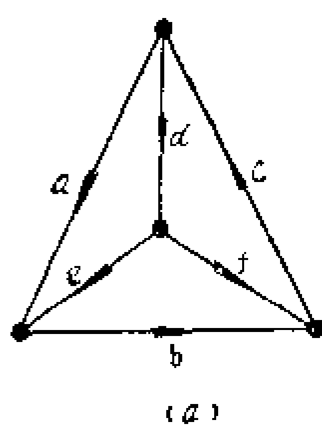
如果两条有向边以同一方向映射到一对顶点时，这两条边叫作平行边。这就是说，除了必须象在无向图中那样地平行外，也要所指的方向一致。在图7—1（a）中，边h，m，n是平行的，而边b和c都不算平行的。

一个有向图有它对应的无向图。反过来，设有一个无向图G，我们可以把各边任意指定方向，这样结果的有向图，有时标以$\vec{G}$叫作与G相伴的有向图，或叫作G的定向。这里要注意，一个给定的有向图只有一个唯一的（在同构性意义上）对应它的无向图，而一个给定的无向图却可能有许多个定向。这就是为什么说对应某向图的特定无向图，而却说相伴某个无向图的一个有向图（可有多个）。

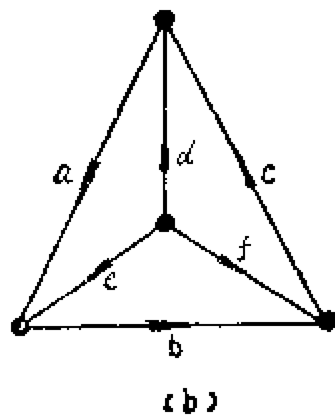
那么什么叫同构有向图呢？所谓同构图是有恒等的特点。换句话讲，如果它们的标号删除，两个同构图是没有区别的。而两个有向图要是同构的话，那么不但它们对应的两个无向图必须是同构，而且对应边的方向也必须一致。例如，图7—2中的两个有向图不是同构的，因为虽然它们所对应的两个无向图是完全相同的，但是两图中的边d的方向正相反。

由于有向图每边所规定的方向不同，有向图的类型较它们相伴的无向图为多。

简单有向图：一个既无自环又无平行边的有向图叫作简单有向图。例如，图7—2和图7—3。



（a）



（b）

图 7—2

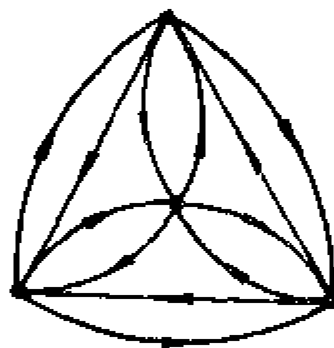


图 7—3

非对称有向图：有向图中一对点之间最多只有一条有向边，但允许有自环的，叫作非对称有向图。

对称有向图：有向图中有某一个方向（如从点a到点b）的有向边，就也有另一个方向（如从点b到点a）的有向边的，叫作对称有向图，例如图7—3。

一个有向图既简单又对称的，叫作简单对称有向图，一个有向图既简单又非对称的，叫作简单非对称有向图。

完全有向图：一个简单有向图其中从每个点指向每个其它点只有一条有向边的，叫作完全对称有向图，例如图7—3。一个非对称有向图如果每一对点间只有一条边的，叫作完全非对称有向图，例如图7—2。

一个有n_v个点的完全非对称有向图含有$n_v(n_v-1)/2$条边，但是一个有n_v个点的完全对称有向图却含有$n_v(n_v-1)$条边。一个完全非对称有向图也叫作一个比赛图或完全比赛图。这种图将在下面讲到。

平衡有向图：一个有向图其中每一个点v_i的入次数等于出次数的，即$d^+(v_i)=d^-(v_i)$，叫作平衡有向图，也叫作伪对称有向图。如果一个平衡有向图的每个点有相同的入次数和出次数的，则叫作正则平衡有向图。

第二节　有向图的关联矩阵

与一个有向图相关的矩阵是与第五章中所讨论过的无向图的有关矩阵十分相似的。不过它有一个基本区别处，即由于各边的定向，有向图的各种矩阵的元素有3个值，即+1，0，和－1，而无向图的矩阵只有0和1。这些值+1，0，和－1，看作是普通的实数，即在实域中的。所以加法和乘法是用普通算术，而不像无向图中那样用模2算术。

一个有向图中各条边的位置和定向的一种表示法是用一个关联矩阵。一个有n_v个点，n_e条边，和没有自环的有向图的关联矩阵A是一个$n_v\times n_e$的矩阵，其中行对应着点，列对应着边，定义如下：

$$A=[a_{ij}] \qquad (7-3)$$

式中　$a_{ij}=1$，如果边j自点i射出；

$a_{ij}=-1$，如果边j从点i射入；

$a_{ij}=0$，如果边j并不关联着点i。

图7—4是一个线性图G。它的关联矩阵为

$$A(G)=\begin{array}{c} \\ 1\\2\\3\\4\\5\\6\end{array}\begin{array}{c}\begin{array}{cccccccc} a & b & c & d & e & f & g & h\end{array}\\ \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\\ -1 & -1 & -1 & 0 & -1 & 0 & 0 & 0\\ 0 & 0 & 1 & 1 & 0 & 0 & -1 & 0\\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}\end{array} \qquad (7-4)$$

如果不管各条边的定向而相应地把式（7—4）中的－1改为1，那么式（7—4）就变成图5—1的关联矩阵式（5—33）了。

在第五章，第二节中所讲的从一个无向图的关联矩阵中所观察到的6个信息，只要稍作修改，也适用于这里的有向图。

在定理5—2中一个关联矩阵的秩数等于它的无向图的秩数。这对于有向图也是适用的。

定理7—1：一个有向图关联矩阵的秩数是n_v-k，其中n_v是有向图的点数，k是部分数。

从关联矩阵A中移去任何一行，得$(n_v-1)\times n_e$的缩减关联矩阵A_f。对应移去那一行的点叫作参考点。

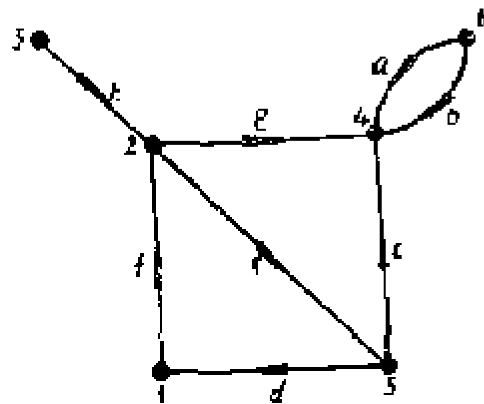


图 7—4

在一个无向图的关联矩阵A中，A的每个方子矩阵的行列式是0或1，这是因为行列式是用模2算术演算的，它的结果只能是0或1，而不会有别的值。

对于有向图呢？关联矩阵A是在实域中。初看起来，方子矩阵的行列式可能得到任何整数值。下列这个定理告诉我们不是那样一回事。

定理7—2：一个有向图的关联矩阵A中每个方子矩阵的行列式的值是1，−1，或0。

证：这个定理的证明可以把矩阵A的一个方子矩阵直接展开而得到的。假定A的一个$m\times m$方子矩阵M。如果M有任何一行或列的元素值都是零，那么很明显M的行列式$|M|=0$。同样，如果M的每一列含有两个非零元素值，一个1和一个−1，那么$|M|=0$。

现在，如果$|M|\neq 0$，那么M中每行的元素值之和不能是零。所以M一定有一列其中有一个单独非零元素值，或是+1，或是−1。设这个单独元素值是在M的(i, j)处。因此

$$|M|=\pm 1|M_{ij}| \qquad (7—5)$$

其中M_{ij}是M删除去行i和列j后的子矩阵。因为M是非奇异的(就是$|M|\neq 0$)，所以$(m-1)\times(m-1)$的子矩阵M_{ij}也是非奇异的。所以M_{ij}一定也有一列其中有一个单独非零元素值，譬如说在(p, q)处。对于(p, q)处的这个元素展开$|M_{ij}|$。

$|M_{ij}|=\pm|M$的一个非奇异$(m-2)\times(m-2)$子矩阵$|$

重复应用这种步骤最后得到

$$|M|=\pm 1 \qquad (7—6)$$

任何一个矩阵它的每一个方子矩阵的行列式的值为1，−1，或0的叫作一个幺模矩阵。幺模矩阵在线性规划中有重要作用。

第三节 有向子图

一个有向图的子图，如行走，路径和回路等，统称为有向子图，它们除了有所对应的无向图的子图的那些特性外，还要考虑到定向。所以有向子图有些特殊性质是对应的无向子图所没有的。

一个有向行走是一个行走，它的定向是从起点到终点，并且行走中每条边的定向是与整个行走的定向一致的。例如，在图7—5中，$W(a, b, c, d, e)$是一个有向行走，但是$W(a, b, f)$却不是一个有向行走，而只是对应着有向图的一个行走，称为半行走。

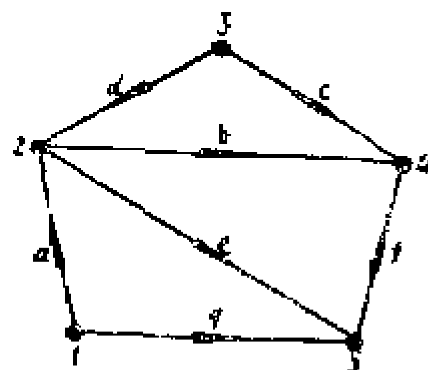


图 7—5

如果一个有向行走的起点和终点是同一点，这个行走就叫作有向闭行走；否则就叫作有向开行走。例如，在图7—5中，$W(b, c, d)$是一个有向闭行走，而(a, b, c, d, e)是一个有向开行走。

在有向图中，前面已经讲过，一个点v_i的次数分为出次数$d^+(v_i)$和入次数$d^-(v_i)$；可是一个点v_i的总次数是与无向图中点v_i的次数相同的，即

$$d(v_i)=d^+(v_i)+d^-(v_i) \qquad (7-7)$$

在一个行走中，虽然一条边不能出现一次以上，一个点却是允许出现多次的。但是在一个开行走中，如果没有一个点出现一次以上，这个开行走就叫做路径。在有向图中，一个有向路径的定义除了与无向图中的路径一样外，还要求由起点到终点有一个定向，而且路径中每条边的定向是与路径的定向一致的。例如，在图7－5中，由起点1到终点3之间的W（a，e，f，c）是一个有向路径，面行走（g，f，b，d）却不是一个有向路径而只是一个半路径。在有向图中，如只提“路径”两字时，那就是指有向路径或半路径。同样，在有向图中提到“行走”时，就指有向行走或半行走。

有向回路和半回路也可以用同样方法下定义。在图7－5中，边组（b，c，d）是一个有向回路，而边组（a，e，g）只是一个半回路，两者都是回路。象在无向图中一样，有向图中两个回路（有向回路或半回路）的环和或者是另一个回路或者是边不共用回路的并集。但是两个有向回路的环和不一定是另一个有向回路或边不共用有向回路的并集。例如图7－5中，有向回路（b，c，d）和（c，d，e，f）的环和，即

$$(b,\ c,\ d)\oplus(c,\ d,\ e,\ f)=(b,\ e,\ f)$$

而（b，e，f）不是一个有向回路而是一个半回路。

在第二章中，我们说过一个图（指无向图）中在一对点之间至少有一个路径的就叫作连通图。在有向图中，既然有两种类型的路径，有向图的连通度就也有两种。如果在一个有向图G中，从每一个点到其它任何一个点之间至少有一个有向路径的，这个有向图G就叫作强连通的。如果一个有向图G的对应无向图是连通的，而G不是强连通的，这个有向图G就叫作弱连通的。例如，在图7－2中，图（a）是强连通的，面图（b）是弱连通的。所以，当我们说到一个连通有向图G时，我们只是指对应的无向图是连通的，而这个G可能是强连通的，也可能是弱连通的。

因为在一个有向图G中有两种连通度，部分的连通度就跟着也有两种。在G的部分中有强连通的子图叫作G的片段。例如，图7－6的有向图G有两个部分g_1和g_2，g_1中含有3个片段（a，b），（e，f，g，h），和（n），而边c，d，和m并不包括在g_1的任何片段中。在部分g_2中存在一个片段（p，q，r）。在一个有向图G中，把每个强连通的片段代以一个点（即由一个点代表一个片段），由一个强连通片段到另一个强连通片段之间所有有向边用单一的有向边来代替（即由一条有向边来连接表示两个片段的两个点），这样就成为一个有向图的凝聚G_c。例如，图7－6中的有向图G可以凝聚成如图7－7所示。

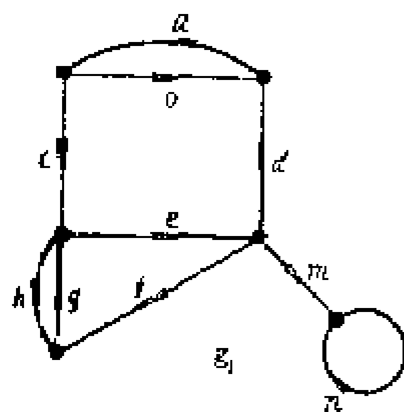


图 7－6

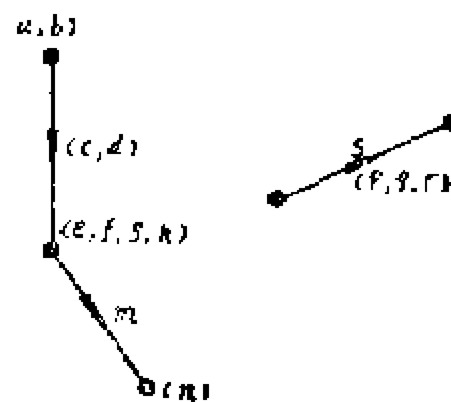


图 7－7

由凝聚的定义可以看出：

（一）一个强连通的有向图只凝聚成一个点。

（二）一个凝聚了的有向图是没有有向回路的。

在一个有向图G中，如果从点i到点j有一个有向路径，我们就说从点i可达点j。这种性质称为可达性。很明显，当且仅当一个有向图G中的每一个点可以从其它任何一个点到达时，这个G是强连通的。图7—2（a）就是这种有向图的一个例子。

第四节　有向欧拉图

欧拉图的概念可以推广到有向图。在一个有向图G中，一个有向闭行走经过G的每一条边正好只有一次的，叫作有向欧拉线。含有一条有向欧拉线的有向图叫作有向欧拉图。有向欧拉图也可以定义为一个有向图其中每一个点的出次数$d^+(v)$等于入次数$d^-(v)$。例如，图7—8（a）是一个有向欧拉图，而图（b）却不是有向欧拉图而是一个欧拉图或是连通有向图。

定理7—3：当且仅当一个连通有向图G中有一条含有G中所有边的有向闭行走时，这个有向图G是一个有向欧拉图。

例如，图7—8（a）的有向欧拉图是含有这个图中所有各边的一个有向闭行走（a，b，c，d，e，f，g，h）。

图　7—8

当一个连通有向欧拉图的每个点的次数为2的（就是说，只有两条边关联着每个点），这就是一个有向回路。例如，图7—8（a）中有下列各有向回路：（a，b，c，d，h），（a，b，c，g，h）（a，d，f，h），（a，f，g，h），（b，c，d，e），（b，c，e，g），（d，e，f），和（e，f，g）。

如果C_1和C_2是两个有向回路，而$C_1\cap C_2\neq\phi$，那么$C_1\oplus C_2$不是一个有向回路。一般地讲，两个有向欧拉图的环和可能不是一个有向欧拉图。

定理7—4：设E是一个有向欧拉图，设E_1是E的一个真子图，并且也是一个有向欧拉图。那么

$$E\oplus E_1=E\ominus E_1 \qquad (7—8)$$

是一个有向欧拉图。

这个定理可以考虑$E\ominus E_1$中各个点的入次数和出次数来证明的。

定理7—5：一个有向欧拉图至少含有一个有向回路。

这个定理可以考虑建立一个有向闭行走来证明。

定理7—6：一个有向欧拉图可以是一个有向回路，或是若干个有向回路的一个边不共用并

集。

证：根据定理7—5，我们知道一个有向欧拉图E至少含有一个有向回路。设这个有向回路为C_1。根据定理7—4，$E\ominus C_1$是一个有向欧拉图，那么在$E\ominus C_1$中至少含有一个有向回路。这样推论下去，最后$E\ominus C_1\ominus C_2\cdots\ominus C_k$就空了。所以我们得到一个结论，$E$是有向回路$C_1$，$C_2$，…，$C_k$的一个边不共用并集，当$k=1$时，$E$就显然只是一个有向回路。▮

例如，图7—8（a）的有向欧拉图可以作为两个有向回路（a，d，f，h）和（b，c，e，g）的一个边不共用并集。

在无向图中，我们研究过M-图，它是一个路径和若干回路的一个边不共用并集。在有向图中，也有M（$i\times j$）型的有向M-图，它满足下列三条件：

（一）除点i和j外，对于其它各点$d^+(v)=d^-(v)$；

（二）$d^+(i)=d^-(i)+1$；

（三）$d^-(j)=d^+(j)+1$。

例如，图7—9就是一个M（$i\times j$）型的有向M-图，它完全满足上述3个条件。

当一个M（$i\times j$）型有向M-图不连通的话，那么其中必有一个部分是M（$i\times j$）型有向M-图，而其余的部分一定是有向欧拉图。例如，图7—10是一个M（$i\times j$）型分离有向M-图，其中部分（a）是一个M（$i\times j$）型有向M-图，而部分（b）和（c）都是有向欧拉图。

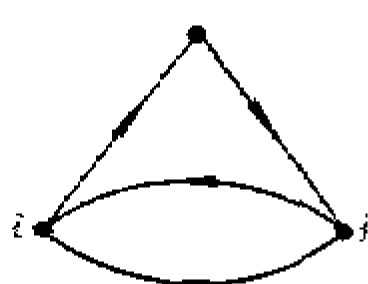
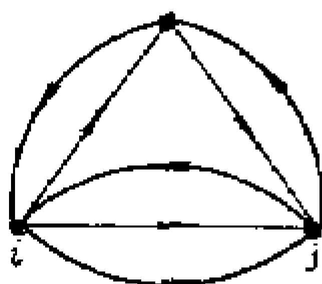
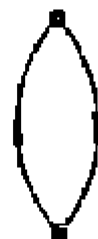


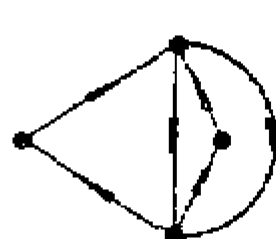
图 7—9



（a）



（b）



（c）

图 7—10

定理7—7：设$M(i\times j)$是一个$M(i\times j)$型有向M-图，并设$M(j\times i)$是一个$M(j\times i)$型有向M-图，假使

$$M(i\times j)\cap M(j\times i)=\phi \qquad (7-9)$$

那么

$$M(i\times j)\oplus M(j\times i)=E \qquad (7-10)$$

是一个有向欧拉图。

这个定理可以用检查E中每一个点的入次数和出次数来证明的。

从上面的定理中可以看出，一个有向欧拉图必须是连通的（可能有的孤立点是除外），否则在一个行走中不可能每条边都经过。事实上，一个有向欧拉图是强连通的。当然，反过来讲，每个强连通的有向图不一定都是有向欧拉图。为了确定一个有向图有无欧拉线，下面这个定理是一个很简单的测试方法。

定理7—8：当且仅当一个有向图G是连通的，并且是平衡的，即G中每一个点的$d^-(v_i)=d^-(v_i)$时，这个图G是一个有向欧拉图。

这个定理的证明，可以参考定理2—1的证明。

第五节 有 向 树

在无向图中的树是一种没有任何回路的连通图。在有向图中，这个概念以及“树这个术语还是适用的。有向图中的一棵树是一个连通有向图，它没有回路（既无有向回路，也无半回路）。有n_v个点的一棵有向树含有n_v-1个有向边，并且它的性质是与无向边的树相似的。有向树在许多应用中是很重要的，诸如在电网络的分析，对策论，语言理论，计算机程序设计，计数问题等等方面。

有向树由于边的定向不同，又可以分为以下两种：

（一）内向树

如果在有向树中存在一个顶点v_0，使树的其它任一顶点v_i都有一条有向路径到达v_0，那么这种有向树称为内向树，如图7—11（a）所示。

（二）外向树

如果在有向树中存在一个顶点v_0，从v_0到任一其它点v_i都有一条有向路径，那么这种有向树称为外向树。v_0的入次数为零。v_0也叫做根，用R表示。如图7—11（b）所示。

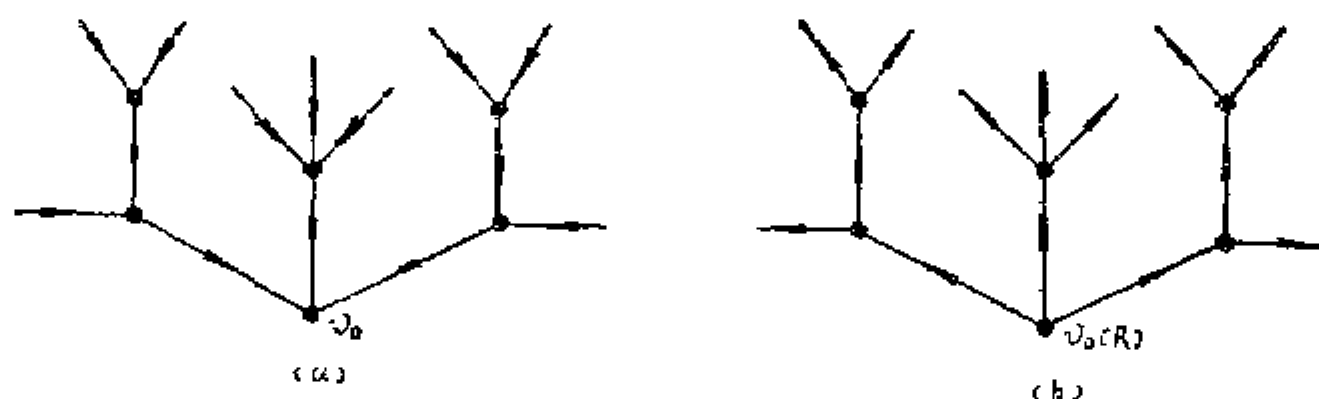


图 7—11

定理7—9：在一个外向树中，除根以外，每一个点的入次数正好为一。

证：一个有n_v个点的外向树G，最多可以有n_v-1条边。所以在G中所有各点的入次数的总和为：

$$d^-(v_1)+d^-(v_2)+\cdots+d^-(v_{n_v})\leqslant n_v-1 \qquad (7—11)$$

式中左侧n_v项中，只有一个点的入次数为零；而其它一定都是正整数，并且一定都是1。现在，既然正好有入次数为1的n_v-1个点以及入次数为零的一个点，有向图G正好有n_v-1条边。既然G没有回路，它一定是连通的，而且是一棵树。▮

定理7—10：在一个外向树中，从根R到每一个其它点有一条有向路径。反过来，如果一个无回路的有向图G中有一个点v，可以从v到达每一个其它点，而从任何其它点不能到达点v，这个有向图G就是一个外向树。

证：（一）在一个外向树中，设想一个有向路径P，以根R沿定向出发，最后只能到达一个悬挂点，否则一个点的入次数会是2或更多。这是矛盾的。因为一个外向树是连通的，每个点一定是在由根R到每一个悬挂点的有向路径之中。

（二）反过来讲，因为G中每一个点可以从点v到达的，而且G没有回路，所以G是一棵树，另外，既然点v不能以任何其它点到达，所以$d^-(v)=0$。每一个其它点可以从点v到达，因此每一个这些点的入次数一定至少为1。因为在G中只有n_v-1条边，其中n_v是G

的点数，所以这些点的入次数不可能大于1。∎

外向树可以应用于计算机的算法理论中。譬如有一个算术表达式：

$$a+b-\frac{cd}{g^x-f} \tag{7—12}$$

在一种程序语言（如象$FORTRAN$或$ALGOL$）中，这个表达式可以写成为

$$a+b-c\cdot d/(g\uparrow x-f) \tag{7—13}$$

其中↑表示取幂。在求这个表达式的值时，计算机必须按照一定次序进行算术运算，否则将算错。让我们把这个表达式中各种运算的次序用数目字标出如下：

$$\underset{⑥}{a+}\ \underset{⑤}{b-}\ \underset{③}{c\cdot}\ \underset{④}{d/}(\underset{①}{g\uparrow}\ \underset{②}{x-}f) \tag{7—14}$$

从这里可以看出，为了求这个表达式的值，计算机必须来回扫描这个表达式，以便找出运算的先后次序。

为了避免计算机的来回扫描，我们在翻译表达式（7—12）时，就可以不用上述语言而用无括号表示法，也称卢柯西维兹表示法。这个表示法有一个优点，就是运算是按照表达式中运算符号出现的先后次序进行的。

无括号表示法的基本想法是把一个二进制算符放在两个运算对象的左侧，而不是象平常那样放在两者之间。例如，$x+y$ 就写成$+xy$。把表达式从程序语言翻译成无括号表示法在编译程序中是极为重要的。这可以先把给出的表达式用外向树表示出来。方法如下：表达式中每一变数（或常数）在外向树中用一个悬挂点表示，每一个内部点连接树干或树枝的点代表一种二进制算符，而把两个子外向树作为它的运算对象。表示出式（7—12），的外向树见图7—12。

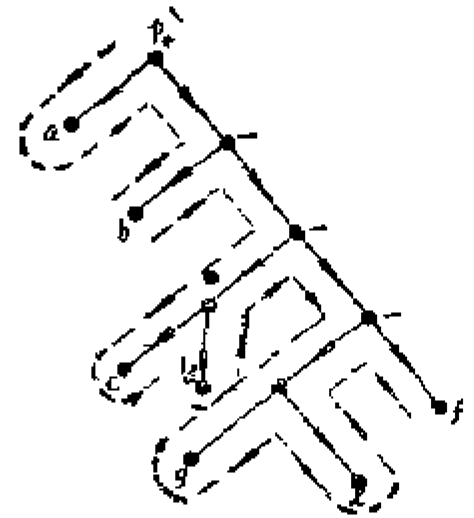


图 7—12

为了得到用无括号表示法翻译式（7—12）的结果，我们就从树根R开始沿着外向树从左到右和从上到下，如图7—12中虚线所示的路线去进行。每次我们碰到以前所未碰到的一个点时，就添上它的标号。在图7—12中进行这个过程的结果是

$$+a-b/\cdot cd-\uparrow gxf \tag{7—15}$$

用无括号表示法来表达式的值的方法是：从式（7—15）的最右方开始向左移动。每当碰到一个算符时，运算总是在紧接这算符右方的两个运算对象中进行。运算所得的结果就作为一个运算对象以备下一次运算用。这样进行下去，我们可以看到式（7—15）所得的结果就等于式（7—14）。式（7—15）比式（7—14）优越的地方在于式（7—15）中没有括号，并且算符按照运算次序从右到左依次出现。所以可以免除运算时计算机的来回扫描。

从图7—12的表达式外向树中可以注意到，各点在纸平面上的相对位置，即左或右，上或下，是很重要的，而且必须保持。在计算机文献中，凡是一棵树的分枝与它相连接的每个点的相对次序必须保持的，叫作有序树，或叫作平面树，后者的命名是因为这棵树可以看作是劲直地埋置于纸平面上。在计算机科学中，树这个术语通常是指有序树，并且按照惯例，一棵树的根总是画在纸面的上部，树是朝下挂着的。

在一个有n_e个点的连通有向图中的一棵生成树是与一个无向图中的一棵生成树相似的，

它包括n_v-1条有向边。一个连通有向图的生成树称为外向生成树。例如，图 7—13 中的（b，d，f）是一棵外向生成树。在一棵外向生成树与一条欧拉线之间有着某些显著的关系，如下面的两个定理所示。

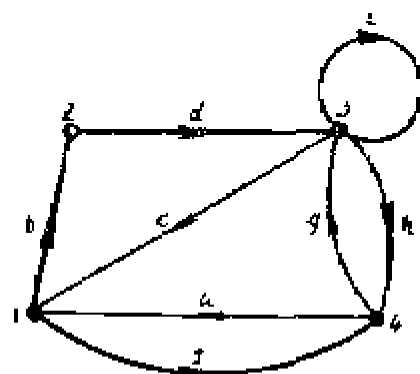


图 7—13

定理 7—11：在一个有n个点和m条边的连通的平衡有向图G中，设$W=(e_1,e_2\cdots,e_m)$是一条欧拉线，它从点v开始又终止于v（就是说点v是边e_1的起点，和边e_m的终点）。在w的m条边中，有$n-1$条边进入除点v以外的每一个$n-1$个点只有一次。由这些$n-1$条有向边连同n个点所组成的有向子图g是G的一个外向生成树，根在点v。

证：在有向子图g中，点v的入次数为零，而每一个其它点的入次数为1。子图g中只包括进入每个$n-1$个点的一条边，而没有进入点v的边。另外，g的形式是由W规定的，它是连通的，并且含有$n-1$条有向边。所以g是G中的一个外向生成树，根在点v。▮

例如，在图 7—13中，$W=(b,d,c,e,f,g,h,a)$是一条欧拉线，由点1开始，又回到点1。有向子图（b，d，f）是一个外向生成树，根在点1。

定理 7—11提供了一个方法，只要一个图是有向欧拉图，就可以得到一个扎根于任何规定点的外向生成树。反过来，给出了一个有向欧拉图中一棵外向生成树，我们可以用下一定理来构成一条欧拉线。

定理 7—12，设G是一个有向欧拉图，和T是G中的一棵内向生成树，根在点R。设e_1是G中的一条边，关联出自点R。如果一个有向行走$W=(e_1,e_2,\cdots,e_m)$是按照下列规则构成的，那么W是一条有向欧拉线。

（一）在W中没有一条边经过一次以上。

（二）从一个点出来，属于T的那条边一直要等到其它外向边都经过后才能用它。

（三）行走只终于那一点。从它那里已经没有余留下任何外向边了。

证：因为图G中所有点一定是进入多少次就出来多少次（因为G是平衡的），行走W一定终止于点R。现在假设在G中有一条边a没有包括在W之中。设点v是边a的一个终点。因为G是平衡的，点v一定也是一条没有包括在W中的边b的起点，否则边b从点v出来，根据规则一，它一定是在T之中。由此推理，又引导到另一条没有包括在W中的边c。这样推论下去，最后到达点R，又找到一条没有包括在W中的外向边。这是与规则（三）相矛盾的。▮

例 7—1　图 7—14中粗线所示是扎根于R的一个内向生成树$T=(b，c，g，m，n)$。从这个内向生成树，我们可以按照定理 7—12所规定的三条规则来得到不同的有向欧拉线。

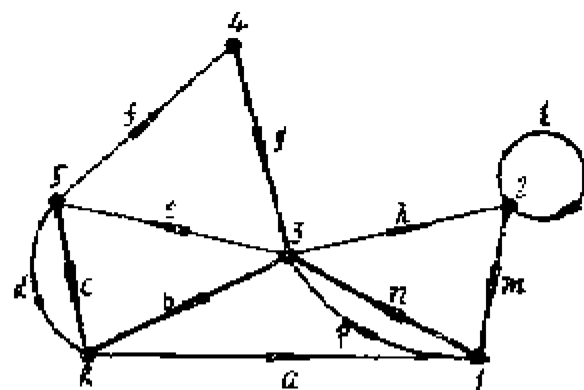


图 7—14

从树根R处的一条边a出发，到达点1后，点1有两条外向边，即n和p。因为边n是属于内向树T的，所以按照规定(二)暂不选用n而先选用边p。到达点3时，这里有3条外向边b，e，和h。因为边b属于T，所以暂不选用b而从边e和h中任选其一。设先选用边e（如先选用边h，则将为另一条有向欧拉线）。到达点5时，这里有两条外向边c和f。我们暂不选用c而先选用边f，到达点4，这里只有一条外向边g，所以没有选择余地而

只能采用它。到达点3时，这里有两条外向边b和h，我们先选用边h。到达点2，这里有两条外向边l和m，我们先选用l。因为边l是条自环，所以又回到点2。这时这里已经没有其它外向边可选，只能用边m。到达点1时，也只能采用边n。到达点3时，只能采用边b。到达点R时，只能采用边d。到达点5时，只能采用边c，又回到了根R。到达根R时，因为外向边a和d已用过，没有任何余留下来的外向边了。所以根据规则（三），有向欧拉线就到此为止。这样选择所得的结果是一条有向欧拉线（a,p,e,f,g,h,l,m,n,b,d,c）。用同样方法，我们可以得另一条有向欧拉线（a,p,h,i,m,n,e,f,g,b,d,c）。

从一个给出有向图中一个给定的内向生成树中某一条边出发究竟能有多少不同的有向欧拉线呢？这可以从树根R的一条边e_1开始，计算每一个点所能有的选择数。到达边e_1的终点v_i时，我们看从这个终点v_i出发，除了包括在内向树T中只有一条外向边，而且根据规则（二），这条边只能等到其它外向边都选用后才能最后选用外，在点v_i处其它可供选用的外向边的选择数为$[d^+(v_i)-1]!$。既然这些都是互不相关的选择，所以总共有

$$\prod_{i=1}^{n}[d^+(v_i)-1]! \qquad (7-16)$$

条不同的欧拉线可以符合定理7—14的三条规则。

根据式（7—16），对图7—14从根R的一条边开始所得的值为2。这个值与例7—1中所得的结果是符合的。注意这里所得的不同有向欧拉线不是有向图G中所有的有向欧拉线数，而只是根据所规定的内向树T而得到的有向欧拉线数。

对于一个有向图，生成树T的符号或是+1或是−1，它就是对应一棵生成树T的A的非奇异子矩阵的行列式值所采用的符号。生成树的符号不是绝对的，而是相对的。这是因为一棵生成树的符号是由A中各点和各边的某一种排列顺序所规定的。如果一个矩阵中的两行或两列互换一下，行列式值的符号就改变了。一棵生成树的符号一经任意选定后，每一棵其它生成树的符号就由于这棵生成树而定为正或负了。

为了确定一个连通有向图中的生成树目的，我们有下面一个定理。

定理7—13：设A_f是一个连通有向图G的缩减关联矩阵。G的生成树数目等于下列行列式值

$$|A_f A_f'| \qquad (7-17)$$

这个行列式的运算系在实域。

证明定理7—13，要用到矩阵代数中的比内-柯西定理。

比内-柯西定理：

设P和Q为两个矩阵，P是$m\times n$阶的，而Q是$n\times m$，且$m\leqslant n$，则

$$det[P、Q]=\sum_i P_i Q_i \qquad (7-18)$$

式中P_i和Q_i是P，Q的m阶行列式，其中P_i中的列对应Q_i中的行，这样的行列式称为对应主行列式

例如：

$$P=\begin{pmatrix}1&0&1\\0&2&2\end{pmatrix} \qquad Q=\begin{pmatrix}1&0\\2&1\\3&0\end{pmatrix}$$

$$det[P、Q]=\begin{vmatrix}1&0\\0&2\end{vmatrix}\begin{vmatrix}1&0\\2&1\end{vmatrix}+\begin{vmatrix}0&1\\2&2\end{vmatrix}\begin{vmatrix}2&1\\3&0\end{vmatrix}+\begin{vmatrix}1&1\\0&2\end{vmatrix}\begin{vmatrix}1&0\\3&0\end{vmatrix}=8$$

下面证明定理 7—13。

因为A_f是 $(n_v-1)\times n_e$阶矩阵，A_f'是$n_e\times(n_v-1)$ 阶，且一般而言 $n_e\geqslant n_v-1$。因为$n_e<n_v-1$时，G是不连通的，因而也就不会有生成树。于是A_f和A_f'满足比内-柯西条件，即

$$det[A_fA_f']=\sum_i A_iA_i=\sum_i A_i^2$$

式中A_i是n_v-1阶行列式，这个行列式的值只可能等于 0 或±1。对于$A_i=0$的那些项在$det[A_fA_f']$中不起作用，而对于$A_i=\pm1$的那些项在 $det[A_fA_f']$中是存在的。可是根据第五章第三节所述，连通图G的一棵生成树对应着A中一个非奇异的主子矩阵，所以

$$\sum A_i^2(A_i\neq 0)=\sum_{树}(\pm1)^2$$

因此，$det[A_fA_f']$=有向树数。∎

无向树的数目也可以应用式（7—17）推导。

同样，有向图G的所有生成树的数目也可以相似地应用定理 5—8 来计算。不过这个次数矩阵中的次数应该是每点的入次数$d^-(i)$，而$i\neq j$时的$-k$中的k应该是图G由点i到点j的边数。这样的次数矩阵称为入次数矩阵。在有向图中，计算有向图中的所有生成树数的定理应该是

定理 7—14：一个没有自环的有向图G中的根r的所有生成树的数目等于从这个图的入次数矩阵D中删除第r行和r列而得到的余子式值。

在无向图中，生成树数的确定也可以应用这个行列式。

第六节　有向图的回路矩阵

在有向图G中的一个回路，象G的对应无向图中的回路一样，不过再用一个箭头来表示一个回路的定向而已。回路的定向完全是任意的。例如，图 7—15的图G中有 4 个回路，即C_1，C_2，C_3和C_4。C_1的定向是顺时针方向，C_2是逆时针方向，C_3是逆时针方向，C_4是顺时针方向。

设G是一个有n_e条边和q个回路（有向回路和半回路）。q个回路中的每一个给予一个任意定向（顺时针方向或逆时针方向）。那么有向图G的回路矩阵B是一个$q\times n_e$的矩阵，定义如下：

$$B=[b_{ij}] \qquad (7-19)$$

式中 $b_{ij}=1$，如果边j在回路i中，并且边的定向与回路的定向一致；

$b_{ij}=-1$，如果边j在回路i中，但是边的定向与回路的定向相反；

$b_{ij}=0$，如果边j不在回路i中。

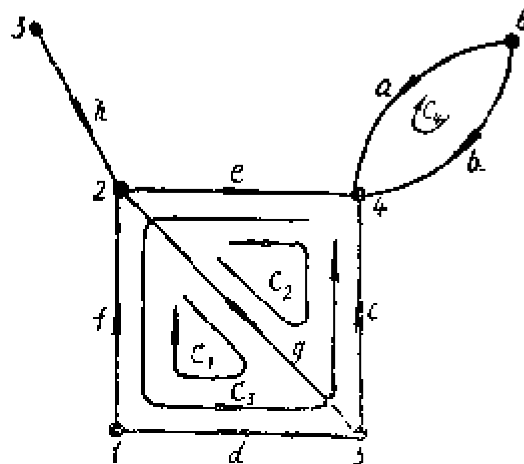


图 7—15

例如，图 7—15中有向图G的回路矩阵为

$$B(G)=\begin{matrix} \\ C_1 \\ C_2 \\ C_3 \\ C_4 \end{matrix}\begin{bmatrix} a & b & c & d & e & f & g & h \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & -1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

在上面的回路矩阵中，如果我们改变任何一个回路的走向，只不过是改变对应行中的每一个非零值的正负号。另外这里也可以注意到，如果我们从行 2 中减去行 1，我们得到行 3。由此可见，各行并不都是线性独立的，但是当然在实域之中。

关于第五章第四节中无向图的回路矩阵的 6 个信息，只要稍作明显的小改变，对于有向图的回路矩阵也是适用的。正象在无向图中那样，回路矩阵的行是正交于关联矩阵的行的（这时在实域中）。

定理 7 —15：设 B 和 A 分别为一个无自环的有向图的回路矩阵和关联矩阵，其中 B 和 A 中各列的排列按照各边的同一顺序，那么

$$AB^t=BA^t=0 \qquad (7-20)$$

其中上标 t 表示转置矩阵。0 表示零矩阵。

证：以 B 的行 m 和 A 中的行 k 为例。如果回路 m 不含关联着点 k 的任何一条边，那么两行的乘积当然是零。另一方面，如果点 k 是在回路 m 之中，就有两条边（譬如说 x 和 y）关联着点 k，并且也都在回路 m 之中。这种情形只有象图 7 —16所示的 4 种方式。（m 的定向相反时，另有 4 种情形是与 x 和 y 互换时的情形一样的）。A 中行 k 和 B 中行 m 在列位置 x 和 y 的可能元素值有 4 种情形如下表所示。在每种情形的点乘积都是零。

图 7 —16

行 k 和行 m 中的可能元素值　　表 7 — 1

情形	行 k		行 m		点乘积
	列 x	列 y	列 x	列 y	行 k · 行 m
(1)	−1	1	1	1	0
(2)	1	−1	−1	−1	0
(3)	−1	−1	1	−1	0
(4)	1	1	−1	1	0

例 7 — 2　以图 7 — 4（b）中的 A 和图 7 —15的 B 为例来检验式（7 —20）。

$$\begin{array}{cc} & \begin{array}{cccccccc} a & b & c & d & e & f & g & h \end{array} \\ A= & \begin{pmatrix} 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & -1 & -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & -1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$\begin{array}{cc} & \begin{array}{cccccccc} a & b & c & d & e & f & g & h \end{array} \\ B= & \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & -1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

所以

$$AB'=\begin{pmatrix} 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & -1 & -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & -1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & -1 & 0 \\ 1 & 0 & -1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}=0$$

这里等式的右项0表示零矩阵。

应用雪尔凡斯探定理和定理7—15可以证明在一个有n_e条边的有向图G中，B的秩数R_B加上A的秩数R_A的和为n_e，即

$$R_A+R_B=n_e \tag{7—21}$$

对于一个连通图来讲

$$R_A=n_v-1 \tag{7—22}$$

所以

$$R_B=n_e-n_v+1 \tag{7—23}$$

矩阵A和B的下列两个重要特性对于有向图也是适用的，不过这里不用模2算术而用通常的实域算术。

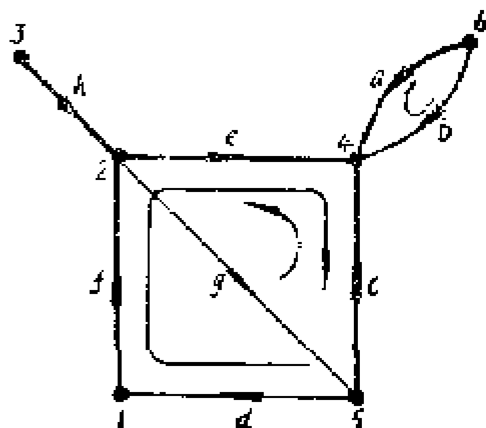
（一）A的那些n_v-1阶非奇异子矩阵是一对一地对应于有n_v个点的连通有向图的那些生成树的。

（二）N阶（$=n_e-n_v+1$）的B的那些非奇异子矩阵是一对一地对应于有n_v个点和n_e条边的连通有向图的弦组（即生成树的余图）。

一个有向图的一个基本回路，也象无向图一样，是由某一特定的生成树和一根弦所组成的。所以每一个基本回路的定向是选择与弦的方向一致的。基本回路数等于图的圈数N。基本回路矩阵B_f的方式，与一个无向图的B_f一样，是一个$N\times n_e$矩阵。

$$B_f=[U_N \vdots B_t] \qquad (7—24)$$

其中U_N是N阶的单位矩阵，B_t的各列对应着生成树中各边。图 7 —17中的粗线是一棵生成树，是对应这棵生成树的基本回路矩阵。



$$B_f=\begin{pmatrix} b & d & g & a & c & e & f & h \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & -1 & 0 & 0 \end{pmatrix}$$

图 7—17

第七节 有向图的截割组矩阵

设V_1和V_2分别是一个无向图的点组V中的两个真子组，符号 $E(V_1\times V_2)$ 表示一个边组，其中每一条边是由V_1中的一个点连接到V_2中的一个点的。在一个无向图中，由于各边没有定向，所以

$$E(V_1\times V_2)=E(V_2\times V_1) \qquad (7—25)$$

可是，在一个有向图中

$$E(V_1\times V_2)\neq E(V_2\times V_1) \qquad (7—26)$$

例如，在图 7 —18中，如果$V_1=(1,2)$，$V_2=(3,4)$，并且$V_2=\overline{V}_1$，其中$\overline{V}_1$代表一个图中点组V中除V_1以外的所有点组，即

$$V_1\cup\overline{V}_1=V \qquad (7—27)$$

和

$$V_1\cap\overline{V}_1=\phi \qquad (7—28)$$

那么图 7 —18中 $E(V_1\times\overline{V}_1)$ 是 (b,d,e)，而$E(\overline{V}_1\times V_1)$ 则由于边的定向关系而是空的。

假使在一个线性图G中，我们删除一个截割组S中的所有各边，我们就把G分离为两个连通子图g_1和g_2。设V_1是g_1中所有点的点组，和$\overline{V}_1$是g_2中所有点的点组。如果G是一个无向图，那么 $E(V_1\times\overline{V}_1)$ 就是截割组S。可是，如果G是有向图，$E(V_1\times\overline{V}_1)$ 就不一定是截割组S了。在一个有向图中，一个截割组S可以表达为

$$S=E(V_1\times\overline{V}_1)\cup E(\overline{V}_1\times V_1) \qquad (7—29)$$

所以在式（7—29）的截割组S，当$E(V_1\times\overline{V}_1)$和 $E(\overline{V}_1\times V_1)$ 中的一个（但不是两个）是空组的话，这个S就叫作有向截割组。

因为在有向图中边有定向，截割组也有定向。在截割组 $S=E(V_1\times\overline{V}_1)\cup E(\overline{V}_1\times V_1)$ 中，S的定向或是由V_1到$\overline{V}_1$或是由$\overline{V}_1$到V_1。为了方便起见，用一个箭头来表示一个截割组的定向。例如，在图 7 —19中，截割组S的箭头表示S的定向是从V_1到$\overline{V}_1$的，其中V_1是g_1中所有各点的点组，$\overline{V}_1$是g_2中所有各点的点组，即使 $S=E(V_1\times\overline{V}_1)\cup E(\overline{V}_1\times V_1)$ 是截割组的一个边不共用并集，对S的定向也只给一个方向，或者从V_1到$\overline{V}_1$，或者从$\overline{V}_1$到V_1。

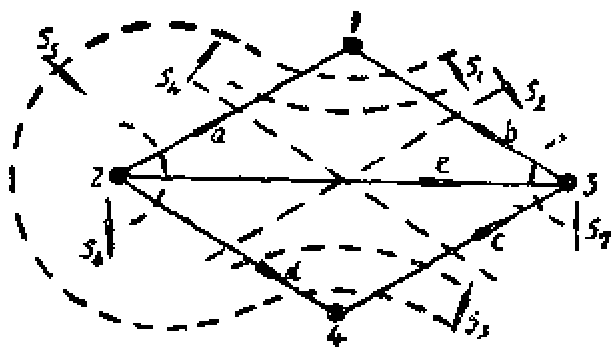


图 7—18

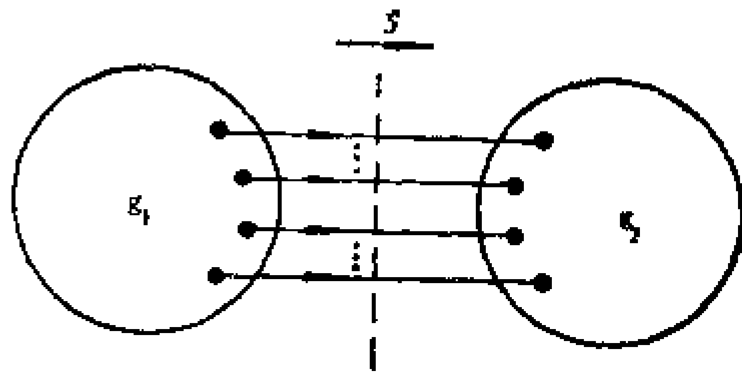


图 7—19

我们可以不用V_1和$\overline{V}_1$来表示截割组(或截割组的一个边不共用并集)S的定向，而用两个点p到q来表示定向，当然$p\in V_1$，和$q\in\overline{V}_1$。那时我们就说，截割组S的定向是从点 p 到点 q，这样就意味着点p和点q是分别在被截割组S所分隔开的两个点组之中。

例 7—3　在图 7—18中，这个有向图的截割组以及截割组的边不共用并集是：

$$S_1=E\{(1)\times(\overline{1})\}\cup E\{(\overline{1})\times(1)\}=(a,b)$$
$$S_2=E\{(1,\ 2)\times(\overline{1,\ 2})\}\cup E\{(\overline{1,\ 2})\times(1,\ 2)\}=(b,d,e)$$
$$S_3=E\{(4)\times(\overline{4})\}\cup E\{(\overline{4})\times(4)\}=(c,d)$$
$$S_4=E\{(1,\ 3)\times(\overline{1,\ 3})\}\cup E\{(\overline{1,\ 3})\times(1,\ 3)\}=(a,c,e)$$
$$S_5=E\{(1,\ 4)\times(\overline{1,\ 4})\}\cup E\{(\overline{1,\ 4})\times(1,\ 4)\}=(a,b,c,d)$$
$$S_6=E\{(2)\times(\overline{2})\}\cup E\{(\overline{2})\times(2)\}=(a,d,e)$$
$$S_7=E\{(3)\times(\overline{3})\}\cup E\{(\overline{3})\times(3)\}=(b,c,e)$$

其中S_1，S_2，S_3，S_4，S_6和S_7是有向截割组，而S_5是截割组的一个边不共用并集。这些截割组的定向选得如图 7—18所示。例如，S_5的定向是从（1，4）到（2，3）。如果认为用点 1 和 2 而不用点组（1，4）和（2，3）更为方便的话，那么我们可以说S_5的定向是从点 1 到点 2 的。

现在我们对一个有向图G的截割组矩阵C定义如下：

$$C=[c_{ij}] \qquad (7—30)$$

式中　$c_{ij}=1$，如果边 j 是在截割组 i 之中，并且边 j 的方向与截割组的方向一致；

$c_{ij}=-1$，如果边 j 是在截割组 i 之中，但是边 j 的方向与截割组的方向不一致；

$c_{ij}=0$，如果边 j 不在截割组 i 之中。这样G中每一个截割组和截割组的边不共用并集将由矩阵C中的一行来代替。

例如，图 7—18中的截割组矩阵C是

$$C=\begin{array}{c} \\ S_1\\ S_2\\ S_3\\ S_4\\ S_5\\ S_6\\ S_7\end{array}\begin{array}{c}\begin{array}{ccccc} a & b & c & d & e\end{array}\\ \left(\begin{array}{ccccc} -1 & 1 & 0 & 0 & 0\\ 0 & -1 & 0 & -1 & -1\\ 0 & 0 & -1 & 1 & 0\\ 1 & 0 & 1 & 0 & 1\\ -1 & 1 & 1 & -1 & 0\\ -1 & 0 & 0 & -1 & -1\\ 0 & -1 & -1 & 0 & -1\end{array}\right)\end{array}$$

一个有向图G的截割组矩阵的秩数是n_v-k，其中n_v是G的点数，k是G的部分数。如果G是一个连通图，$k=1$，秩数就是n_v-1。

一个有n_v个点和n_e条边的连通有向图的基本截割组矩阵G_f也与无向图的C_f一样，是C的一个$(n_v-1)\times n_e$子矩阵，不过另加每个基本截割组的定向条件。每个基本截割组的定向选择得与某棵特定生成树中在这个基本截割组中的树枝的方向一致。用这些基本截割组，我们得到一个基本截割组矩阵C_f为

$$C_f=[C_c \vdots U_{n_v-1}] \tag{7—31}$$

其中C_c是对应着生成树T以外的弦所组成的子矩阵，U_{n_v-1}是一个对应着生成树的树枝的单位矩阵。

例 7—4　我们在图 7—18中选择一棵由边组（a，d，e）所组成的生成树T。这个有向图的基本截割组有

$$S_a=E\{(2,3,4)\times(1)\}\cup E\{(1)\times(2,3,4)\}$$
$$S_d=E\{(1,2,3)\times(4)\}\cup E\{(4)\times(1,2,3)\}$$
$$S_e=E\{(1,2,4)\times(3)\}\cup E\{(3)\times(1,2,4)\}$$

其中S_a的定向为从（2,3,4）到（1），S_d的定向为从（1,2,3）到（4），S_e的定向为从（1,2,4）到（3），所以对应生成树T的基本截割组矩阵C_f是

$$C_f=\begin{matrix} S_a \\ S_d \\ S_e \end{matrix}\overset{\begin{matrix} b & \;c & \;\;a & \;d & \;e \end{matrix}}{\left(\begin{array}{cc:ccc} -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{array}\right)}$$

在有向图中，回路矩阵和截割组矩阵也象无向图中那样有下列关系。

定理 7—16：设B和C分别是一个有向图的回路矩阵和截割组矩阵，则

$$BC'=CB'=0 \tag{7—32}$$

在式（7—32）中，如果用同一棵生成树（或林）的基本回路矩阵和基本截割组矩阵，并根据式（7—24）和（7—31），则

$$C_fB_f'=[C_c \vdots U][U \vdots B_t]'=[C_c \vdots U]\begin{pmatrix} U \\ \cdots \\ B_t' \end{pmatrix}=0 \tag{7—33}$$

所以

$$C_c=-B_t' \tag{7—34}$$

或

$$C_f=[-B_t' \vdots U] \tag{7—35}$$

同样，基本回路矩阵B_f可以表达为

$$B_f=[U \vdots -C_c'] \tag{7—36}$$

例如，在图 7—18的有向图中，对于生成树$T=(a,d,e)$的基本截割组矩阵C_f有

$$C_c=\begin{pmatrix} b & c \\ -1 & 0 \\ 0 & -1 \\ 1 & 1 \end{pmatrix}$$

所以有向图中对于生成树T的基本回路矩阵B_f为

$$B_f=[U \,\vdots\, -C_{c}^{t}]=\begin{array}{c} \\ \end{array}\overset{\begin{array}{ccccc} b & c & a & d & e \end{array}}{\left[\begin{array}{cc:ccc} 1 & 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & 1 & -1 \end{array}\right]}$$

第八节 有向图的邻接矩阵

设G是一个有n_o个点但没有平行边的有向图。G的邻接矩阵X是一个$n_o\times n_o$的（0，1）方矩阵，它的定义是

$$X=[x_{ij}] \qquad (7-37)$$

式中$x_{ij}=1$，如果有一条边出自点i指向点j；

$x_{ij}=0$，如果没有边出自点i指向点j。

图7—20是一个有向图。它的邻接矩阵为

$$X=\begin{array}{c} \\ 1\\2\\3\\4\\5 \end{array}\overset{\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array}}{\left[\begin{array}{ccccc} 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{array}\right]} \qquad (7-38)$$

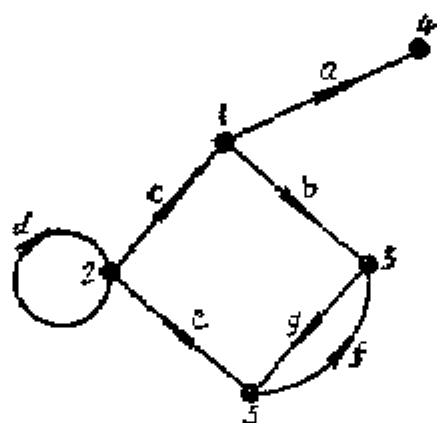


图 7—20

邻接矩阵在各种学科中有不同的名称。在时序机理论中叫作转移矩阵，在关系学法中叫作关系矩阵，在网络流中叫作接通矩阵或优先矩阵，在调度和关键路径分析法中叫作前趋矩阵。

一个有向图G的邻接矩阵X有下列特性：

（一）当且仅当G是一个对称有向图时，X是一个对称矩阵。

（二）在主对角线上每一个非零元素值代表着对应点上的一个自环。

（三）在X中无法显示出平行边。这就是为什么在邻接矩阵的定义中规定没有平行边。

（四）每行的和等于对应点的出次数，每列的和等于对应点的入次数。X中非零值的数目等于G中的边数。

（五）任何两行的置换连同相应列的置换不改变有向图，置换只是意味着各点的重新编号。所以当且仅当两个有向图的邻接矩阵置换后能恒等时，它们是同构的。

（六）如果X是一个有向图G的邻接矩阵，那么转置矩阵X'是把G中每条边的方向颠倒后所得到的那个有向图G^R的邻接矩阵。

（七）对于n阶的任何（0，1）方矩阵Q，存在着一个有n个点的唯一有向图G，它的邻接矩阵就是Q。

邻接矩阵是用来作为研究有向图性质的工具，尤其是在用数字计算机时更为有用。

关于一个有向图G的邻接矩阵的幂，这里有一个定理与定理5—7有相似之处。

定理7—17：在矩阵X^r的(i, j)处的元素值等于从点i到点j之间r条边的不同有向边序列数。

证：这个定理可以用归纳法来证明。对于$r=1$，这个定理是正确的。作为归纳假设，假定这个定理对于X^{r-1}也是适用的。根据归纳假设，在$X^r(=X^{r-1}X)$的(i, j)处的元素

值

$$x_{ij}^{r}=\sum_{k=1}^{n}[\text{在}X^{r-1}\text{中的}(i,\ k)\text{处元素值}]\cdot x_{kj}$$

$$=\sum_{k=1}^{n}(\text{由点}i\text{到}k,\ \text{段数为}r-1\text{的所有有向边序列数})\cdot x_{kj} \qquad (7-39)$$

在式（7—39）中，根据从点k到点j有无一条有向边，而$x_{kj}=1$或0。因此当且仅当从i到j有一个段数为r的有向边序列，而它的最后一条边是从k到j时，式（7—39）的和中有一项不是零。如这项不是零，它的值等于从i经k到j的这种边序列数。这对于每一点k，$1\leqslant k\leqslant n$都是适用的。所以，式（7—39）等于从i到j所有可能的有向边序列数。

这里应注意，X^r中$(i,\ j)$处元素值给出从点i到j的所有有向边序列数，这些边序列有三类：

（一）从i到j的有向路径：这些有向边序列中，没有一个点是通过一次以上的。

（二）从i到j的有向行走：这些有向边序列中，一个点可能通过一次以上，但是没有一条边是通过一次以上的。

（三）这些有向边序列中，一条边也可能通过一次以上的。

不幸的是没有一个简单方法来区分这三类。这就是为什么这个简单方法不能用来计算某些特定段数的有向路径或有向回路。

例如，检查图7—20中有向图的邻接矩阵的4次幂X^4

$$X^4=\begin{pmatrix}1&1&0&0&1\\1&2&2&1&1\\1&2&2&1&0\\0&0&0&0&0\\2&2&1&1&2\end{pmatrix}$$

行2和列3处的元素值代表两个有4段的有向边序列：c,b,g,f（由点2到3的有向行走）和d,d,c,b（不是一个行走）。第3对角元素值代表从点3开始又终于点3的两个有4段的有向边序列：g,e,c,b（一个有向回路）和g,f,g,f（不是一个有向回路）。X^4中的其它元素值可用同样的方法检查。

对于一个有n_0个点的简单有向图G，一个$n_0\times n_0$矩阵叫作克希荷夫矩阵或K-矩阵。K的定义如下：

$$K(G)=[k_{ij}] \qquad (7-40)$$

式中$k_{ii}=d^-(v_i)$，点i的入次数。

$k_{ij}=-x_{ij}$，邻接矩阵中$(i,\ j)$处带负号的元素值（此处的元素值系指由点i到j的有向边数）。

例如，图7—21是一个简单有向图，它的K-矩阵为

$$K(G)=\begin{array}{c}\\1\\2\\3\\4\end{array}\begin{array}{c}\begin{array}{cccc}1&2&3&4\end{array}\\\begin{pmatrix}1&0&0&-1\\-1&2&-1&0\\0&-1&2&-1\\0&-1&-1&2\end{pmatrix}\end{array} \qquad (7-41)$$

在一个K-矩阵中，每列各元素值之和等于零。这表示n。行是线性相关的。所以K-矩阵的行列式值是零，即

$$|K| = 0 \tag{7—42}$$

定理 7—18： 设$K(G)$是一个简单有向图G的K-矩阵。$K(G)$的(q, q)余因子值等于G中扎根于点v_q的外向树数。

例如，在图 7—21中，在K-矩阵行 2 中，每个元素值的余因子是 3，这表示这个有向图有 3 个外向树扎根于点 2。

对于一个有向欧拉图G，因为每行和每列的和都是零，$K(G)$的所有余因子都是相等的。设$K(G)$的所有余因子的公值是σ。这个σ就是G中扎根于任何一个给定点的不同外向树数。式（7—16）给出结合这些不同外向树中每一个的不同欧拉线数。所以得到下一定理。

定理 7—19： 每一个有向欧拉图的欧拉线数为

$$\sigma \cdot \prod_{i=1}^{n} [d^+(v_i) - 1]! \tag{7—43}$$

从这一定理我们可以计算在任何一个连通的平衡的有向图中的欧拉线数。例如，试计算图 7—22的欧拉线数。它的K-矩阵是根据有向汉密尔顿路径b, c, d, k, l, n, o顺序中出现的各点排列的。

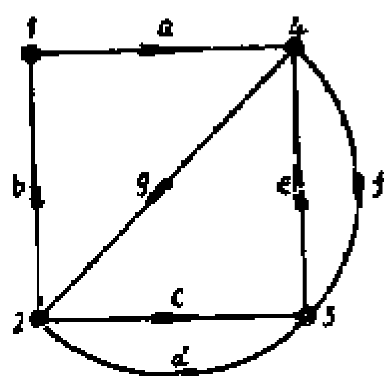


图 7—21

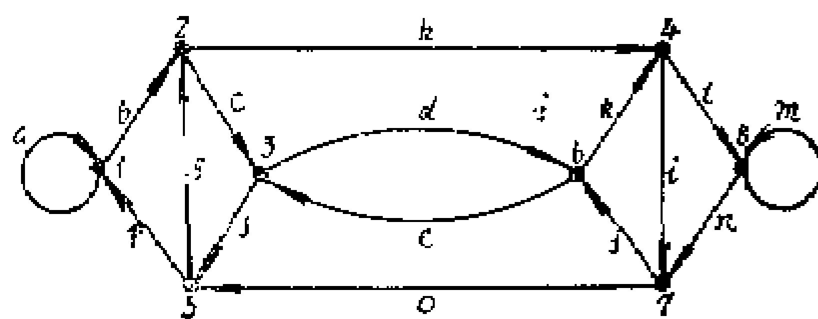


图 7—22

$$K(G) = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 2 & -1 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 2 & -1 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

在这个矩阵中，每项的余因子是16，所以式（7—43）中的σ是16。因为图 7—22中每个点的v_i的$d^+(v_i) = 2$，所以

$$\prod_{i=1}^{8} [d^+(v_i) - 1]! = 1$$

因此，图 7—22中的欧拉线数是16。

第九节 成对比较和比赛图

在很多情况下，尤其在社会科学方面，人们要求在若干给出的事物中，每次只对两个作一比较，然后对这若干事物评出高低优劣，排出它们的等级或名次。这种方法叫作成对比较法。这种方法用在难于用数量来衡量的情况。例如，对音乐作品的个人爱好，一般是每次对两个作品欣赏一回，评定个人的爱好。把 n 个作品组成各种可能的 $n(n-1)/2$ 对后，相互比较，最后把 n 个作品评出一个高低名次。

有向图是适合于成对比较法结果的一种自然表示方式。曾有人研究过用有向图来表示一只狗对 6 种不同饲料的喜爱等级。图 7 —23中的 6 个点代表 6 种不同的饲料。每天他用两种不同饲料来喂狗，看狗先吃完哪种饲料来判定狗对饲料的喜爱程度。他进行了15天，使可能有的每两种不同饲料的搭配都能试到。在图 7 —23中，每一边的箭头方向是从一种较喜吃的饲料指向较次的饲料。例如，饲料 1 比 2 要较喜欢吃些。这种图叫作优先图。

一般地讲，从一个优先图来评定等级名次不是一件容易的事。例如在图 7 —23中，由于狗性的前后矛盾，这只狗欢喜 1 胜于 2 ， 2 胜于 4 ，但 4 却胜于 1 。所以从这 3 种饲料中，很难确定哪一种是狗最喜爱吃的。

同样的情况也发生在运动比赛中。在一次循环比赛中，一个运动员对其他每一个运动员都比赛过。比赛的结果也可以用一个有向图来表示，一条从点 a 到点 b 的边的方向表示运动员 a 胜过运动员 b 。这就是为什么我们在本章第一节中把一个完全非对称有向图叫作一个比赛图或完全比赛图的理由。所以图 7 —23也可以看作是 6 个运动员的比赛图。在一次比赛中运动员的名次的排定问题是与成对比较法中的评比问题一样的。

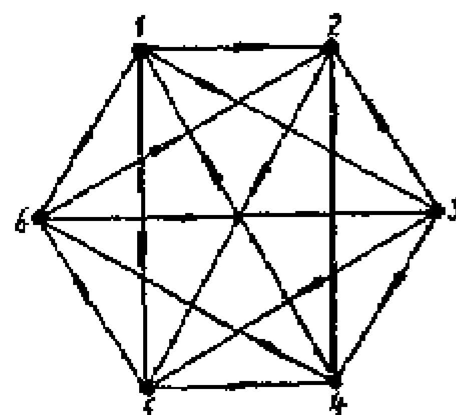


图 7 —23

在循环比赛中，一种惯常使用的直截了当的评定方法是根据每一个运动员的得分记录，就是说一个运动员的获胜次数。在上面所举狗饲料的例子中，一只狗对某一种饲料较喜爱的次数就相当于得分记录。在一次比赛中，一个运动员的得分次数等于一个有向图中对应点的出次数。

如果我们用得分记录来评定名次，那么在上述的狗饲料中，我们会排出 6 种饲料的优先次序如下：

（1，3），（2，5，6），和 4。

就是说，饲料 1 和 3 是同为第一等，饲料 2， 5， 6 同为第二等，饲料 4 是最不喜爱的，评为第三等。

根据有向图中每个点的出次数来评定等级名次是一个最容易的方法，但是不一定总是一个满意的办法。特别是，如果比赛不完全，譬如说，运动员们不是竞赛同样的局数，这个方法就失去了意义。

另外一种有时用来评定运动员优胜名次的方法是应用有向汉密尔顿路径，就是每一个运动员战胜了他的后继对手。在图 7 —23中，这样的一个名次是 1 — 3 — 2 — 5 — 6 — 4 。关于比赛图中有关汉密尔顿路径有下列一个定理。

定理 7 —20： 每一个完全比赛图有一条有向汉密尔顿路径。

证：这个定理可以在点数方面用归纳法来证明。靠实际画图，这个定理可以证明对有 1，2，3，和 4 个点的完全比赛图是适用的。让我们先作一个归纳假设，就是说这个定理对于有n_0个点的所有完全比赛图是正确的，进而证明对于有 n_0+1 个点的所有比赛图也是适用的。

设G是一个有n_0+1个点的完全比赛图。设g是G中一个n_0个点的完全子比赛图。由于归纳假设，g有一条有向汉密尔顿路径。设这条路径是 1，2，…，n_0。设在g中没有而在G中有的那个点叫作点n_0+1。

因为G是一个有n_0+1个点的完全比赛图，在G中的点 n_0+1 有一条有向边或者进入或者出自其它点 1，2，…，n_0中的一点。那么就有下列 3 种可能情况：

情况一：在点 n_0+1 和点 1 之间的边是指向点 1 的。那么在 G 中有一条汉密尔顿路径（n_0-1，1，2，…，n_0）如图 7—24(a)所示。这就证明了这个定理。

情况二：在点n_0和点n_0+1之间有一条边指向点n_0+1。那么我们在G中又有一条有向汉密尔顿路径（1，2，…，n_0，n_0+1，如图 7—24(b)所示。这样又证明了这个定理。

情况三：要不是上面这两种情况，而是这两条边中的一条是从点 1 指向n_0+1，而另一条从点n_0+1指向点n_0的。在这种情况下，当我们从点 1 移向点n_0时，我们碰到关联着点n_0+1的两条边中出现了一个方向颠倒。我们把那第一次出现这种方向颠倒的点叫作点 i（点 i 也可能就是点n_0本身）。这种方向颠倒一定是因为边（i，n_0+1）是指向点n_0+1的，但是边（n_0+1，$i+1$）是出自点n_0+1的。在这种情况时，我们在G中有一条有向汉密尔顿路径（1，2，…，i，n_0+1，$i+1$，…，n_0）如图 7—24(c)所示。这种就证明了这个定理的正确性。

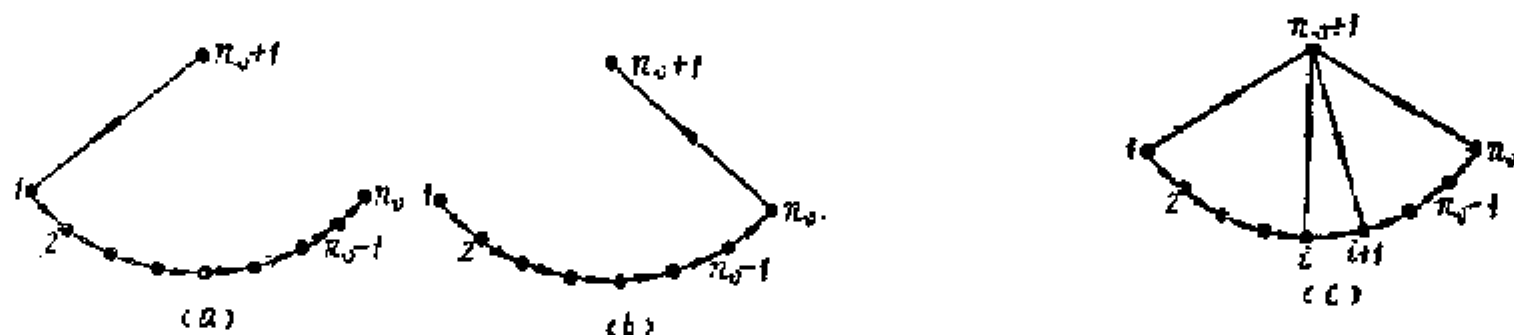


图 7—24

所以，如果一个有向图是一个完全比赛图，其中至少总可能有一个汉密尔顿评比法。可是这种评比法也有些缺点的，其一，这种评比法与运动员得分之间有些不符。其二，一个比赛图可能有一个以上的有向汉密顿路径，因此有几个不同的评比法。例如，在图 7—23中，1—3—2—5—6—4 和 1—3—5—6—2—4 是两种不同的汉密尔顿评比法。

在任何一个比赛图（完全的或不完全的）中，对于n_0个点的评比，如果点 j 领先于点 i，一次违章是作为一条由点 i 到 j 的边的。例如，在图 7—23中，名次 1—3—2—5—6—4 有下列两次违章，即由点 4 到 1 和由点 6 到 2 的两条有向边。而名次 3—2—5—6—4—1 有下列五次违章，即由点 1 到 3，由点 1 到 2，由点 6 到 2，由点 1 到 5，和由点 1 到 6 的五条有向边。

违章次数最少的评比法代表某一次比赛的最少可能的混乱。其实，这种评比法自动地包括得分评比法和汉密尔顿评比法。所以这种方法可以算得是评比法中最好的方法。

可是要从有 n 个点的所有 $n!$ 个可能名次中找出其中违章次数最少的一个，在计算上也

是困难的。曾有人用过一种用动态规划的方法来评比，到目前为止它是一种可以采用的最好方法，不过计算起来比较慢，而且麻烦。

在所有$n!$个名次中违章次数最少的方法代表着有向图中的一个最小边组，它的删除将切断所有的有向回路，而使有向图成为非循环（或零调）有向图，这个问题将在下一节中讨论。

第十节　非循环有向图和消除循环法

在很多情况下，人们只注意着一个给定的有向图中是否有有向回路，而对于半回路是无关紧要的，一个没有有向回路的有向图叫作非循环有向图或零调有向图。非循环有向图有下列特性：

（一）每一棵含有向边的树是一个非循环有向图，但是反过来讲是不正确的。例如，图7—25是一个非循环有向图，但不是树。

（二）一个非循环有向图不能凝聚。这就是说，一个非循环有向图G的凝聚图G_c就是G本身。反过来讲也是正确的，因为如果$G_c=G$，很明显G是不含有向回路的。

（三）当且仅当在一个有向图G中每一个有向行走也是一条有向路径时，G是非循环的。

（四）上面的第三个特性有一个重要含意：如果一个有向图是非循环的，在X^k的(i, j)处的元素值给出由点i到点j之间段数为k的不同有向路径数。

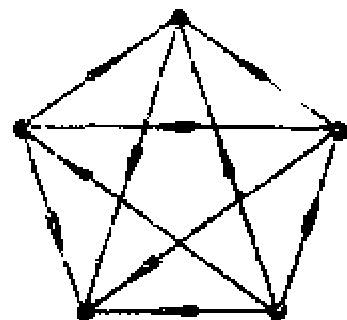


图 7—25

下面我们举出几条有关非循环有向图的定理。

定理7—21：每一个非循环有向图G至少有一个点入次数为零，以及至少有一个点的出次数为零。

证：我们先考虑G中任何一条最长的有向路径P。所谓最长的路径是指一条路径，它的段数不能由于在任何一端再增加一条边而增加的。设点v是路径P的起点，w是它的终点。因为G是非循环的，v和w一定是分离的两点。现在G中各点可以分成两组：点组V_1是在P上，而点组V_2是其余诸点。

关联到点v的没有从V_1中任何点来的边，否则G会有一个有向回路了。同样，也没有从V_2中任何点来的边关联到点v，否则P的段数会由于增加了这条边而增加了。所以点v的入次数为$d^-(v)=0$。同样，因为没有边关联出自点w，所以$d^+(w)=0$。

定理7—22：当且仅当一个有向图G的各点可以按这样的次序编排，使它的邻接矩阵X是一个上（或下）三角矩阵时，G是非循环的。

证：（一）假设X是一个上三角矩阵，就是

$$\text{对于 } i\geqslant j, \quad x_{ij}=0 \qquad (7—44)$$

我们知道由于直接相乘，X^2也是一个上三角矩阵，且X的其它所有幂X^3，X^4，…也都是上三角矩阵。因为X的所有幂的每个对角线处的元素值都是零，所以G没有有向回路，这就是说G是非循环的。

（二）为了证明定理的第二部分，假设G是非循环的，并且把G的各点另行编排如下，根据定理7—21，G中至少有一个点的入次数为零。在重新编排各点时，让这个点为第一个

点v_1。现在删除这个点v_1以及关联到v_1的各边，因为G是非循环的，所以余留下来的有向图$G\ominus v_1$一定也是非循环的。因此$G\ominus v_1$中也至少有一个点的入次数为零。设点v_2就是这样的一个点，并且在重新编排时作为第二个点v_2。次从$G\ominus v_1$中删除这个点v_2。继续这样进行下去，把G中各点重新编排为v_1，v_2，…。

现在考虑G的邻接矩阵X，它的各点已如上述重新编排，列1（对应v_1）的元素值都是零。在行1下面的列2代表$G\ominus v_1$中的点v_2，也都是零。余类推。这样，这个邻接矩阵是个上三角矩阵。这就证明了定理的第二部分。

（三）定理中的下三角矩阵部分也可以用同样重新编排有零出次数的各点来证明的，或者考虑X^t和G^R来证明的。∎

为了检查一个有向图是否是非循环的，可应用下面的一个定理。

定理7—23：当且仅当一个有向图G的行列式$|U-X|$不等于零，其中U是与X同阶的单位矩阵时，G是非循环的。

证：当且仅当$[U-X]^{-1}$存在时，矩阵$[U-X]$才是非奇异的，即$|U-X|\neq 0$。但是

$$[U-X]^{-1}=U+X+X^2+\cdots+X^n+\cdots \tag{7—45}$$

当且仅当无穷级数式（7—45）收敛时，这个逆$[U-X]^{-1}$才存在；就是说，对于所有$k\geqslant N$，$X^k=0$（因为X只含有非负元素值）。可是，当且仅当G不含有段数为N或更多的有向边序列时，对于所有$k\geqslant N$，X^k才是零。这只有当且仅当G不含有任何段数的回路时才是可能的。∎

非循环有向图在很多应用方面是十分重要的。上面讲到过有向回路代表着成对比较评比法中的矛盾性。在电网络中，有向回路可能代表不希望有的反馈电路。在关键路径法或计划评审技术中的计划图中，一个有向回路代表一个严重错误，并且必须去除的。这是因为一个有向回路，譬如$a\ b\ c\ a$暗示着活动a必须在活动b之前完成，b在c之前完成，和c又在a之前完成。很明显，这是一种不可能的情况，要是这样，就什么也做不成。在计算机程序设计中也常常发生一种与此相似的情况，有人称这种情况为致命遭遇是有理由的。

在演绎逻辑中（在那点代表着公理或陈述。有向边代表着定理或从一个陈述引向另一个陈述的推导），一个有向回路意味着兜圈子论证，所以是一种谬论。

所以知道如何用最小的努力来破除这些有缺点的回路是很重要的。换句话讲，如何从一个给出的有向图G中删除一个最小的有向边组，使G变为非循环。所以我们要研究消除循环法。

例7—5　图7—2(a)中有下列几个有向回路（a，b，c），（c，d，f），和（b，c，d，e）。在这个简单的例子中，一检查就可以看出删除边c就可以消除所有的有向回路。这条边c就是最小的边组，删除后使余留的有向图成为非循环的了。

这样一个最小边组删除后能够破坏一个有向图G中所有的有向回路，这个边组在电工学中叫作最小反馈弧组。一般地讲，一个有向图可以有几个最小反馈弧组。从一个有向图中得到这样一个最少边组的方法叫作一个有向图的最小消除循环法。

一个任意的有向图中作最小消除循环的工作是一桩单调而重复的工作，到目前为止，还没有找到一个简单方法。有一个方法是应用定理7—22如下：尽可能多地置换邻接矩阵X中的行和对应的列作出上三角矩阵。在主对角线上和以下所余留下来的那些1代表着一个最小反馈弧组。

另外一个方法是：

（一）在一个给出的有向图G中，求得所有的有向回路。

（二）把每一个有向回路用它那些边的布尔和表达出来。

（三）在第二步中所得到的所有的有向回路表达式中取布尔乘积，在这里应用布尔代数中的吸收律，如$a\cdot a=a$，$a+a=a$，和$a+ab=a$等。

（四）乘积的和中每项代表一个边组，它的删除将破坏所有的有向回路。选取边数最少的那项，那就是最小反馈弧组。

例7—6　在图7—2(a)中，所有的有向回路为：(a，b，c)，(c，d，f)和(b，c，d，e)。将这3个回路先用布尔和方式表达出来，然后把这3个布尔和乘出来，最后化简，得

$$(a+b+c)(c+d+f)(b+c+d+e)=ad+aef+bd+bf+c$$

很清楚，等式右边的每一项代表着一个边组，它的删除会破坏图7—2(a)中的所有有向回路。边数最少的一组是$\{c\}$，这就是我们所要找的最小反馈弧组。

上述两种方法对于大的有向图就不实用了。

习　题

7—1　证明任意一个有向图的所有点的入次数之和等于它们的出次数之和，并等于有向图的边数。

7—2　画出1点、2点和3点所有不同的（非同构的）的简单有向图。

7—3　写出图7—26的关联矩阵。

7—4　写出图7—26对某一棵生成树的基本回路矩阵和基本截割组矩阵。

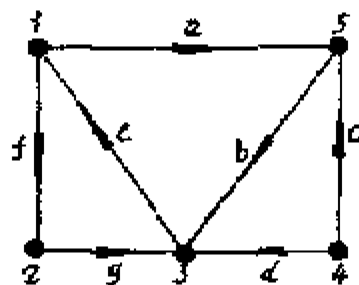


图　7—26

7—5　证明当且仅当定义为

$$M=X+X^2+\cdots+X^n$$

的矩阵M具有非零元素时，一个n个点的有向图是强连通的，式中X是邻接矩阵。

7—6　证明有向图中每条边都属于一个有向回路或一个有向截割组。

7—7　利用习题7—4实例，说明有向图的基本回路矩阵和基本截割组矩阵的转置矩阵的乘积为零。

7—8　利用有向图求解下列古典问题：

"有3个吃人的野人和3个可供野人吃的传教士需要渡河，一只最多只能载两人的小船，3个传教士和1个野人会划船。但是不论何时何岸，野人的数目不能超过传教士的数目，否则传教士就会被吃掉。"（提示：用点代表不同的状态，用有向边表示可能的过渡）。

7—9　一个渡船工要把一头狼、一只山羊和一篮白菜带过河去。因为他的船小，他只能带其中的一样。狼不允许单独与山羊在一起，山羊也不许和白菜单独在一起。问渡船工必须怎样渡河？

第八章　电网络的拓扑分析

第一节　引　　言

网络拓扑是拓扑学的一个分支，是图论在网络分析中的应用。网络的范围很广，本章只讨论电网络的拓扑分析。图论原是数学的一个分支，它在电网络方面的应用是从十九世纪中期开始的。1847年克希荷夫用图论解决电路问题，利用树的概念建立电路方程。1892年麦克斯韦发展了克希荷夫的结论，创立了克希荷夫第三、第四方程。可是当时运算工具还十分简单，靠手算只能分析一些小网络，因此利用图论来分析小网络并不比网络的一般分析法显得优越，这就使得克希荷夫他那超越时代的发现当时未能受到重视。

近年来，由于高速电子计算机的出现，利用图论来分析网络已经成为可能，于是人们对图论的研究兴趣重新高涨起来。现在图论已经渗透到许多学科领域，当然也大量地应用于电网络。图论为网络分析与综合开创了崭新的途径，成为网络理论研究的一个不可缺少的数学基础。

电网络是由一些电路元件（或器件）按照一定的规律连接起来的一个集体。电网络的特性是由下面两个因素决定的：第一是组成网络的元件特性；第二是网络的拓扑结构。目前电网络中所用的元件大致有以下 4 种分类法：

（一）集中参数与分布参数；

（二）线性元件与非线性元件；

（三）时变元件与定常元件；

（四）无源元件与有源元件。

本章主要讨论线性、定常、无源和集中参数电路的拓扑分析法，也对有源网络的拓扑分析作必要的说明。分布参数电路可以利用流图的方法处理，有兴趣的读者可以查阅有关专门书籍。

图论在网络中的应用常常沿用网络分析中的术语，例如网络N中的节点在线图中称为点，网络N中的支路在线图中称为边等等。一般不会发生误解，必要时再加以说明。

第二节　电网络方程

众所周知，克希荷夫定律是网络分析的基本定律，它描述了网络所代表的实际系统的平衡性能。电网络分析通常用电流和电压作为变量。著名的克希荷夫电流定律和克希荷夫电压定律，描述了网络电流和电压的平衡关系。

克希荷夫电流定律（简写为KCL）：对于任一集中参数的电路中的任一节点，在任一时刻，流入（或流出）节点的所有支路电流的代数和等于零。

KCL可以由节点推广到任一闭合面，即通过闭合面的所有支路电流的代数和等于零。

克希荷夫电压定律（简写为KVL）：对于任一集中参数的电路中的任一回路，在任一

时刻，沿着该回路的所有支路电压的代数和等于零。

上述这两个定律只与网络的结构有关，与支路中具体接什么元件无关。所以只研究网络的拓扑性质，就可以用线段来代替元件，得到一个表示网络拓扑结构的线性图，或称拓扑图，简称为图。网络拓扑分析就是将网络的拓扑特性和元件特性分别考虑，利用拓扑公式求解。下面就分别讨论网络的拓扑特性方程式和元件特性方程式，统称为电网络的基本方程式。

电网络基本方程式包括三个内容：

（一）KCL方程；

（二）KVL方程；

（三）电压电流关系方程，简称VCR方程。VCR方程就是指支路电流和电压间的关系式。若支路中只有一个元件，实际就是指元件上电压和电流的关系式，即元件上的欧姆定律。

上述方程中前两个表示网络的拓扑特性，而后者表示支路特性。

在图8—1所示的电网络中，含有6个支路，4个节点，每个支路的电流和电压都有一个参考方向。假设电流参考方向如图8—1中箭头所示，而电压采用相联参考方向，则网络的拓扑性质可用图8—2所示的线性图来进行分析。

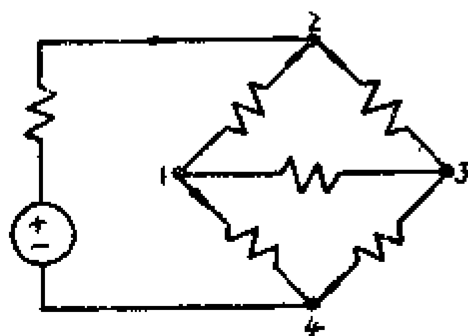


图 8—1

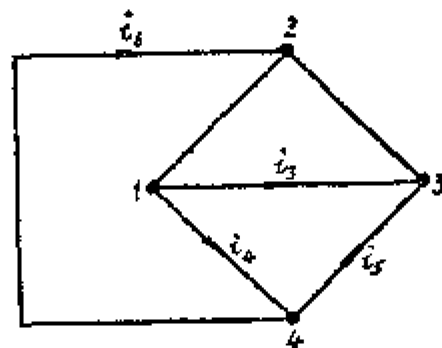


图 8—2

图8—2是一个有向图，每条边上的边权为i_k，它的方向与网络中支路电流的参考方向一致。这个有向图就表示电流有向图。由于电压与电流采用相联参考方向，所以图8—2中的边权若为v_k，则它又表示电压有向图。由此可以看出，在无源网络分析中，电流和电压有向图是不必加以区分的，所以都笼统地称为网络的线性图。

设支路电流的列矢量为：

$$I=[i_1, i_2, i_3, i_4, i_5, i_6]'$$

图8—2的关联矩阵A为

$$A=\begin{matrix} \\ 1 \\ 2 \\ 3 \\ 4 \end{matrix}\begin{matrix} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{bmatrix} -1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 1 \end{bmatrix} \end{matrix}$$

用A左乘I，得到图8—2中4个节点上的KCL方程，因此必恒等于零。即

$$A\cdot I=\begin{bmatrix} -i_1+i_3+i_4 \\ i_1+i_2-i_6 \\ -i_2-i_3-i_5 \\ -i_4+i_5+i_6 \end{bmatrix}=0$$

对于有n_v个节点，n_e条支路的一般情况来说。

$$I=[i_1,i_2,\cdots,i_{n_e}]^t \qquad (8-1)$$

KCL方程为

$$A\cdot I=0 \qquad (8-2)$$

显然，由于A矩阵的秩$R(A)=n_v-1$，所以KCL的独立方程为：

$$A_f\cdot I=0 \qquad (8-3)$$

式中A_f是缩减关联矩阵。

当然，KCL还可以推广到闭合面，若闭合面将图分成两个独立部分，则通过闭合面的支路构成一个截割组。如果图的截割组矩阵为C，则有

$$C\cdot I=0 \qquad (8-4)$$

显然式（8－4）也并非每行都是独立的。截割组矩阵C的秩$R(C)=n_v-1$，所以C中只有n_v-1行是独立的。一般选取基本截割组作为独立截割组，所以有：

$$C_f\cdot I=0 \qquad (8-5)$$

式中C_f为基本截割组矩阵。

在第五章中曾经介绍过，一个图的基本截割组矩阵C_f可以通过缩减关联矩阵A_f的非奇异变换而求得。所以式（8－5）可以由式（8－3）推出。设A_f和C_f的列都对应某一棵生成树的弦和枝，按照先弦后枝的顺序排列，则有：

$$A_f=[A_c \vdots A_t]$$

$$C_f=[C_c \vdots U_t]$$

和

$$C_c=A_t^{-1}\cdot A_c$$

式中　U_t是n_v-1阶单位矩阵。

于是式（8－3）可以写成：

$$[A_c \vdots A_t]\cdot I=0$$

因为A_t是非奇异的，所以其逆矩阵存在，用A_t的逆A_t^{-1}乘上式得：

$$A_t^{-1}[A_c \vdots A_t]I=0$$

即得

$$C_f\cdot I=0$$

方程式（8－3）和（8－5）称为KCL的矩阵形式。

若图8－2以电压作为边权。设支路电压的列向量为：

$$V=[v_1,v_2,\cdots v_{n_e}]^t$$

则用图的回路矩阵B左乘V，根据KVL应恒等于零。

即

$$B\cdot V=0 \qquad (8-6)$$

回路矩阵B的一行对应一个回路，其中每个元素描述了回路与边的关系。若第i行元素为$b_{i1},b_{i2},\cdots,b_{in_e}$，其中非零元素就表示该元素所在的列对应的那条边与回路关联，而零元

素所在的列对应的那条边与 i 回路不关联。因此 B 中第 i 行元素乘相应的支路电压，就是沿 i 回路的 KVL，这就必恒等于零。式（8—6）称为克希荷夫电压定律的矩阵形式。

图8—2的 KVL 方程为：

$$
B\cdot V=\begin{matrix} \\ b_1\\ b_2\\ b_3\\ b_4\\ b_5\\ b_6\\ b_7\end{matrix}\begin{matrix}\begin{matrix}1 & 2 & 3 & 4 & 5 & 6\end{matrix}\\ \begin{bmatrix}-1 & 1 & -1 & 0 & 0 & 0\\ 0 & 0 & 1 & -1 & -1 & 0\\ 1 & 0 & 0 & 1 & 0 & 1\\ -1 & 1 & 0 & -1 & -1 & 0\\ 0 & 1 & -1 & 1 & 0 & 1\\ 1 & 0 & 1 & 0 & -1 & 1\\ 0 & 1 & 0 & 0 & -1 & 1\end{bmatrix}\end{matrix}\begin{bmatrix}v_1\\ v_2\\ v_3\\ v_4\\ v_5\\ v_6\end{bmatrix}
$$

$$
=\begin{bmatrix}-v_1+v_2-v_3\\ v_3-v_4-v_5\\ v_1+v_4+v_6\\ -v_1+v_2-v_4-v_5\\ v_2-v_3-v_4-v_6\\ v_1+v_3-v_5+v_6\\ v_2-v_5+v_6\end{bmatrix}=0
$$

在写回路矩阵 B 之前，应先选择回路的参考方向，上式的回路参考方向是顺时针的。

回路矩阵 B 的秩，$R(B)=n_e-n_v+1$，所以 B 中只有 n_e-n_v+1 行是独立的。如何选取独立回路呢？一般在平面网络中可以选取网孔，而通常就取基本回路，于是 KVL 的独立方程为：

$$M\cdot V=0\quad 或 B_f\cdot V=0 \tag{8—7}$$

式中 M 是网孔矩阵；B_f 是基本回路矩阵。

必须指出，C_f 和 B_f 是对应某一棵生成树的基本截割组矩阵和基本回路矩阵，它们的方向是可以任意选定的。但是，为了方便起见，一般都采用树支方向作为基本截割组的方向，以弦方向为基本回路方向。

公式（8—3）和（8—7）表示网络的独立电流方程和独立电压方程。由此可见，无论是支路电流或支路电压并不全部是独立的变量，KCL 给出了 n_v-1 个电流约束，KVL 给出了 n_e-n_v+1 个电压约束，两个定律一共给出 n_e 个约束，所以独立支路电流和独立支路电压数只能有 n_e 个。下面讨论独立支路电流变量和独立支路电压变量。

设网络 N 的图为 G，选 G 的一棵生成树 T，按照先弦后枝的顺序写出 A_f，C_f 和 B_f。

即

$$A_f=[A_c \vdots A_t]$$
$$C_f=[C_c \vdots U_t]$$
$$B_f=[U_c \vdots B_t]$$

同样按照先弦电流，后枝电流，先弦电压，后枝电压的顺序写出 I 和 V

$$I=\begin{bmatrix}I_c\\ \cdots\\ I_t\end{bmatrix}$$

$$V=\begin{bmatrix} V_c \\ \cdots \\ V_t \end{bmatrix}$$

式中　I_c、V_c表示弦电流和弦电压，

　　I_t、V_t表示枝电流和枝电压。

根据式（8－3）有：

$$[A_c \vdots A_t]\cdot\begin{bmatrix} I_c \\ \cdots \\ I_t \end{bmatrix}=0$$

$$A_c\cdot I_c+A_t\cdot I_t=0$$

因为A_t是非奇异的，

所以
$$I_t=-A_t^{-1}\cdot A_c I_c$$

$$I=\begin{bmatrix} I_c \\ \cdots \\ I_t \end{bmatrix}=\begin{bmatrix} I_c \\ \cdots\cdots \\ -A_t^{-1}A_c I_c \end{bmatrix}=\begin{bmatrix} U_c \\ \cdots\cdots \\ -A_t^{-1}A_c \end{bmatrix}I_c=B_f^t I_c \qquad (8-8)$$

由上式可见，n_e个支路电流并非都是独立的，其中树枝电流I_t可以用弦电流I_c来表示，全部支路电流也就可以用I_c来表示，所以弦电流I_c是独立的电流变量。

同理，根据式（8－7）

$$[U_c \vdots B_t]\begin{bmatrix} V_c \\ \cdots \\ V_t \end{bmatrix}=0$$

$$V_c=-B_t V_t$$

所以

$$V=\begin{bmatrix} V_c \\ \cdots \\ V_t \end{bmatrix}=\begin{bmatrix} -B_t V_t \\ \cdots\cdots \\ V_t \end{bmatrix}=\begin{bmatrix} -B_t \\ \cdots \\ V_t \end{bmatrix}V_t=C_f^t\cdot V_t \qquad (8-9)$$

式（8－9）表明，n_e个支路电压可以由n_v-1个树枝电压V_t表示，所以树枝电压V_t是独立的电压变量。

电网络方程的第三个内容是支路特性，即VCR。支路特性是与具体支路中元件性质相关联的。为了列出支路电压和电流关系，首先必须阐明电网络中的支路和相应线性图中边的关系。换句话说，如何根据给定的一个电网络作出相应的线性图。对于某一个含线性R，L，C定常元件的网络，首先引入一般支路的概念。一般支路用图8－3表示。下标k表示这条支路在网络图中是第k条支路。z_k是k支路的阻抗，v_{sk}是k支路的独立电压源，i_{sk}是k支路的独立电流源。若k支路是由单个独立电压源v_{sk}或单个无源元件z_k构成时，则只需将一般支路中其它元件用零代替就可以了。顺便指出，$i_{sk}=0$表示电流源开路，而$v_{sk}=0$表示电压源短路。所以图8－3不失其普遍意义，称作一般支路。电网络中的一条一般支路对应着线性图中一条由v_i到v_j的有向边。这样就可以根据线性图引出KCL和KVL的方程。为了避免出现微分方程，独立源和阻抗需先通过拉氏变换，于是一般支路中电流、电压、阻抗的符号均表示对应函数的象函数和运算阻抗。在元件的初始条件为零时，k支路的电压电流关系为：

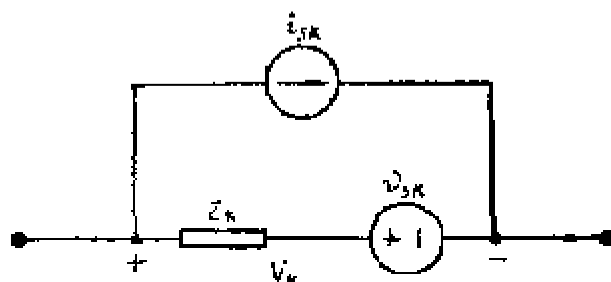


图 8—3

$$v_k = z_k(i_k - i_{sk}) - v_{sk}$$

支路电压，电流关系的矩阵形式为：

$$V = z(I - I_s) - V_s \quad (8—10)$$

式中 V ——支路电压列向量；

I ——支路电流列向量；

Z ——支路阻抗矩阵。

假使网络含有n_e条一般支路，每一支路的运算阻抗为z_k，$k=1,2,\cdots,n_e$。

则

$$Z = diag[Z_1, Z_2, \cdots, Z_k, \cdots, Z_{n_e}]$$

式中 $diag$是对角矩阵的符号。

上面给出的式（8—2），（8—7）和（8—10）称为网络基本方程式。这是网络分析的基本根据。

第三节 节点分析法

由上节可知，网络的支路电压并不全部都是独立的，只有树枝电压是独立的电压变量；支路电流也不全部都是独立的，只有弦电流是独立的电流变量。因此，全部支路电压可用枝电压来表示，而支路电流可以用弦电流来表示。这种电压变量或电流变量的变换过程中首先要选定一棵生成树，再确定一组独立的截割组和独立回路。如果选取的独立截割组正是一组关联组，那么，这种变换就可以不通过树来完成。相应的变换称为节点变换。如所选的独立回路正好是一组网孔，则相应的变换称为网孔变换。上面这些变换导出了网络分析的四种方法：即节点、截割组、回路、网孔。由于推导过程十分类似，本章只以节点分析法为例，说明其拓扑分析。

设网络N对应的线性图为G，G有n_v个节点，于是就有n_v-1个独立关联组。若选v_0为参考节点，其余n_v-1个节点相对应于v_0的电压就称为节点电压，节点电压的列向量记为V_n。我们用G的缩减关联矩阵A_f的转置矩阵A_f^t左乘V_n，其结果正好是支路电压列向量V，即：

$$V = A_f^t \cdot V_n \quad (8—11)$$

这是因为A_f^t的行表示一条边，一条边有二个端点，所以每一行中最多有二个非零元素：一个是$+1$；一个是-1。其中$+1$对应的列号是这条边的起点，而-1对应的列号是这条边的终点。当A_f^t的一行乘以V_n后，就得到这一行对应边的起点节点电压减去终点节点电压，也就是这条边的支路电压。所以式（8—11）是KVL的另一种形式。

分析网络N应当从3个网络基本方程出发，即

（1）KCL： $A_f \cdot I = 0$

（2）KVL： $A_f^t \cdot V_n = V$

（3）VCR： $V = Z(I - I_s) - V_s$

由（3）移项得（4） $I = YV - YV_s + I_s$

式中 $Y = Z^{-1} = diag[y_1, y_2, \cdots, y_k, \cdots, y_{n_e}]$

称为支路导纳矩阵，这里y_k $k=1,2,\cdots,n_e$称为支路运算导纳，$y_k = 1/Z_k$。

据（1） $A_f \cdot I = 0$

所以，A_f左乘（4）即得

$$A_f \cdot I = A_f Y V - A_f Y V_s + A_f I_s = 0$$

将（2）代入上式得

$$A_f Y A_f^t V_n - A_f Y V_s + A_f I_s = 0$$

令 $A_f Y A_f^t = Y_n$

Y_n称为节点导纳矩阵

$$A_f Y V_s - A_f I_s = I_{sn}$$

I_{sn}称为节点电流源向量，则得

$$Y_n V_n = I_{sn} \qquad (8-12)$$

式（8—12）称为节点分析基本方程的矩阵形式，从式（8—12）解出V_n，再代入式（8—11）就得到网络的各支路电压。

求解式（8—12）时，需要用Y_n的逆矩阵Y_n^{-1}，所以Y_n必须是非奇异矩阵才能从下面公式中解出V_n。

$$V_n = Y_n^{-1} I_{sn} \qquad (8-13)$$

为此节点分析法主要应分析Y_n的性质，我们将在下节中专门讨论。在不含互感和受控源网络中，Y是一个对角矩阵，这是显而易见的。因此，Y_n是一个n_v-1阶对称阵。在含互感电路中，Y是一个对称矩阵。在有受控源的网络中，Y则是一个方阵。有源网络的分析将在第八节中专门讨论。

例 8—1　写出图 8—4 的节点分析方程。

因为网络图 8—4 中$n_v=4$，$n_b=5$，它的相应线性图如图 8—5 所示。

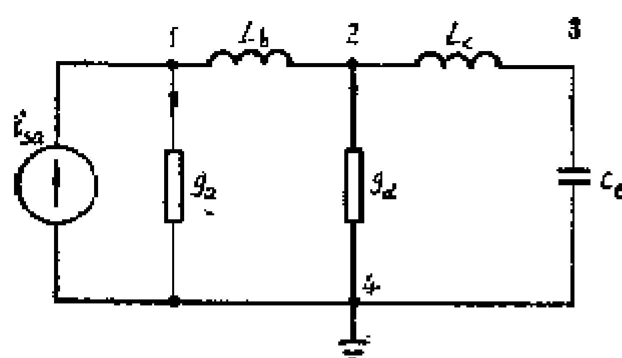


图 8—4

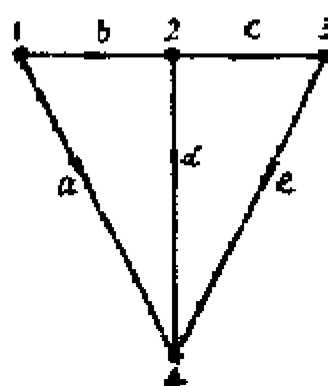


图 8—5

设以点 4 为参考点，支路方向如图中箭头所示，节点分析方程为

$$Y_n V_n = I_{sn}$$

因为$Y_n = A_f Y A_f^t$，所以必须先求A_f和Y，

$$A_f = \begin{matrix} \\ 1 \\ 2 \\ 3 \end{matrix}\begin{bmatrix} a & b & c & d & e \\ 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix}$$

$$Y = diag\left[g_a,\ \frac{1}{SL_b},\ \frac{1}{SL_c},\ g_d,\ SC_e\right]$$

$$Y_n=\begin{pmatrix} g_a+\frac{1}{SL_b} & -\frac{1}{SL_b} & 0 \\ -\frac{1}{SL_b} & \frac{1}{SL_b}+\frac{1}{SL_c}+g_d & -\frac{1}{SL_c} \\ 0 & -\frac{1}{SL_c} & \frac{1}{SL_c}+SC_e \end{pmatrix}$$

$$I_{sn}=[i_{sa},\ 0,\ 0]'$$

所以节点方程为

$$\begin{pmatrix} g_a+\frac{1}{SL_b} & -\frac{1}{SL_b} & 0 \\ -\frac{1}{SL_b} & \frac{1}{SL_b}+\frac{1}{SL_c}+g_d & -\frac{1}{SL_c} \\ 0 & -\frac{1}{SL_c} & \frac{1}{SL_c}+SC_e \end{pmatrix}\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}=\begin{pmatrix} i_{sa} \\ 0 \\ 0 \end{pmatrix}$$

由例中可见，在无受控源和互感情况下，Y_n是一个方阵。这个方阵可以根据网络直接写出，其中主对角线元素就是与该点相联的支路导纳之和，而非对角线元素就是相邻节点间的支路导纳冠以负号。

如果在图 8—4 中，L_b和L_c间有一个互感M存在，假设其对应端如图 8—6 所示，则在写出Y矩阵之前先要写出 b，c 支路的电感矩阵。

因为

$$\begin{pmatrix} v_2 \\ v_3 \end{pmatrix}=S\begin{pmatrix} L_b & M \\ M & L_c \end{pmatrix}\cdot\begin{pmatrix} i_b \\ i_c \end{pmatrix}$$

所以

$$\begin{pmatrix} i_b \\ i_c \end{pmatrix}=\frac{1}{S}\begin{pmatrix} \frac{L_c}{\Delta} & \frac{-M}{\Delta} \\ \frac{-M}{\Delta} & \frac{L_b}{\Delta} \end{pmatrix}\cdot\begin{pmatrix} v_2 \\ v_3 \end{pmatrix}$$

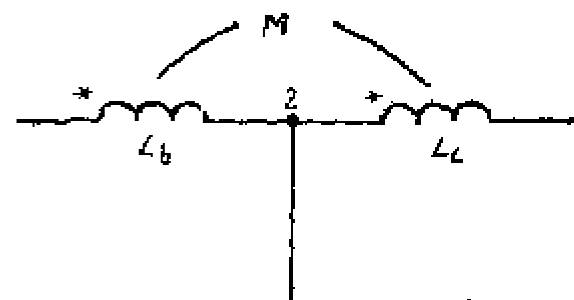


图 8—6

式中　$\Delta=L_bL_c-M^2$

$$Y=\begin{pmatrix} g_a & 0 & 0 & 0 & 0 \\ 0 & \frac{L_c}{S\Delta} & \frac{-M}{S\Delta} & 0 & 0 \\ 0 & \frac{-M}{S\Delta} & \frac{L_b}{S\Delta} & 0 & 0 \\ 0 & 0 & 0 & g_d & 0 \\ 0 & 0 & 0 & 0 & SC_e \end{pmatrix}$$

$$Y_n=\begin{pmatrix} g_a+\frac{L_c}{S\Delta} & -\frac{(L_c+M)}{S\Delta} & \frac{M}{S\Delta} \\ -\frac{(L_c+M)}{S\Delta} & \frac{L_c+L_b-2M}{S\Delta}+g_d & -\frac{(M+L_b)}{S\Delta} \\ \frac{M}{S\Delta} & -\frac{(M+L_b)}{S\Delta} & \frac{L_b}{S\Delta}+SC_e \end{pmatrix}$$

由此可见，节点导纳矩阵就不象无互感时能直接写出了。

第四节　节点导纳矩阵的行列式

上节介绍的节点分析法是根据电网络的基本方程通过节点变换得出的。其中主要应讨论节点导纳矩阵的性质及其计算方法。如果我们把电网络通过广义的KCL经过截割组变换，也可以得出类似节点分析法的截割组分析法。其中也要讨论一个截割组导纳矩阵。它与本节所讨论的节点导纳矩阵是类似的，这虽不加详述。

根据式（8—12）求解V_n时需计算Y_n的逆矩阵Y_n^{-1}。根据线性代数可知

$$Y_n^{-1}=\frac{\overset{*}{Y}_n}{\Delta_n} \qquad (8-14)$$

式中，$\overset{*}{Y}_n$是Y_n的相伴矩阵，Δ_n是Y_n的行列式。

所以

$$\overset{*}{Y}_n=\begin{pmatrix} \Delta_{11} & \Delta_{21} \cdots \Delta_{n_v-1\,1} \\ \Delta_{12} \cdots\cdots & \cdots \Delta_{n_v-1\,2} \\ \cdots\cdots & \cdots\cdots \\ \Delta_{1\,n_v-1} \cdots & \Delta_{n_v-1\,n_v-1} \end{pmatrix}$$

其中Δ_{ij}就是Δ_n中第ij个元素的代数余子式。

定理 8—1：一个包括无源元件的电网络的节点导纳矩阵的行列式等于这个网络所对应的线图的全部生成树的树导纳积的和。记为：

$$\det Y_n=\Delta_n=\sum_{\text{全部生成树}}\text{树导纳积} \qquad (8-15)$$

证：$\det Y_n=\det[A_f Y A_f^t]$

当网络N中不含受控源和互感时，Y是一个对角矩阵，所以令

$$H=A_f Y$$

H是$(n_v-1)\times n_b$阶矩阵，它的形式与A_f相同，每个元素的数值是A_f对应元素乘以该列的支路导纳。

根据比内-柯西定理

有　　$\det HA_f^t=\Sigma H$的主行列式与A_f^t主行列式积

因为H的列对应一棵树的主行列式与A_f^t对应同一棵树的行的行列式乘积等于这棵树的树支导纳乘积，于是上式为：

$$\det Y_n=\sum_{\text{全部树}}\text{树导纳积}。$$

式（8—15）最初是由克希荷夫提出的，之后由麦克斯韦尔总结完成的，所以称为麦克斯韦尔公式，又称克希荷夫第四定律。

例 8—1 中Δ_n可以这样求得，因为图 8—5 的生成树为：abc，abe，acd，ade，ace，bcd，bce，bde。

所以　　$\Delta_n=g_a\frac{1}{SL_b}\frac{1}{SL_c}+g_a\frac{1}{SL_b}SC_e+g_a\frac{1}{SL_c}g_d+g_a\frac{1}{SL_e}SC_e$

$$+g_a g_d SC_e+\frac{1}{SL_b}\frac{1}{SL_c}g_d+\frac{1}{SL_b}\frac{1}{SL_c}SC_e+\frac{1}{SL_b}g_d SC_e$$

显然，这样求Δ_n比从Y_n的行列式计算要方便些，因为它不必计算对消项。

式8—15称为Δ_n的拓扑公式。

顺便指出，回路阻抗矩阵的行列式也有类拟的拓扑公式，读者可参考网络拓扑分析的专著。

第五节 节点导纳矩阵行列式的余子式

计算 $\dot{Y}_n$及网络函数的时候，都要遇到计算节点导纳矩阵行列式的余子式，即Δ_{ii}或Δ_{ij}。如果能直接从网络得到Δ_{ii}和Δ_{ij}，就可以用拓扑方法求Y_n^{-1}或网络函数了。

计算节点导纳矩阵行列式的余子式要比计算Δ_n麻烦得多。为了计算Δ_{ii}和Δ_{ij}，我们先回顾一下Δ_n的计算方法。Δ_n是A_f左乘Y再右乘A_f^t，在A_f矩阵中参考点对应的行是不在内的。根据定义：

$$\Delta_{ij}=(-1)^{i+j}|M_{ij}|$$

式中 M_{ij}是Y_n中删除i行j列后的子矩阵。

于是Δ_{ii}就是从Δ_n中划去第i行和第i列，而Δ_{ij}就是从Δ_n中划去第i行和第j列得到的子行列式。从Δ_n中划去i行就是从A_f中划去第i行，这个矩阵用A_{f-i}表示；而从Δ_n中划去j列就是在A_f^t中划去j列得到的矩阵，可用A_{f-j}^t表示。这样，

$$\Delta_{ii}=\det[A_{f-i}YA_{f-i}^t]$$

$$\Delta_{ij}=(-1)^{i+j}\det[A_{f-i}YA_{f-j}^t]$$

上式中，Δ_{ii}称为对称余子式，而Δ_{ij}称为不对称余子式。

从关联矩阵的定义可以看出，A_{f-i}就是原网络N中将i点和参考点v_0合并后得到的新网络N_1的缩减关联矩阵，而A_{f-j}^t是j和v_0合并得网络N_2的缩减关联矩阵的转置矩阵。于是有

$$\Delta_{ii}=\sum_{N_1\text{的全部树}}N_1\text{的树导纳积}$$

$$\Delta_{ij}=(-1)^{i+j}\sum_{\text{全部公树}}N_1\text{和}N_2\text{的公树导纳积}\tag{8—16}$$

式中N_1和N_2的公树是指这棵树既是N_1的一棵生成树，又是N_2的一棵生成树。式(8—16)是用N_1网络来计算Δ_{ii}和用N_1N_2来计算Δ_{ij}的拓扑公式。这里可以进一步提出，我们能否直接由网络N来计算呢？很明显N是一个n_v个点的图，每棵生成树含有n_v-1条边，而N_1只有n_v-1个点，它的生成树只有n_v-2条边。所以N_1中的生成树是一个含有n_v-2条边且不含回路的连通子图，但这个子图在N中当然也不包含回路，可是却是不连通的。这是因为N有n_v个点，用n_v-2条边是不可能连通n_v个点的。又因为N_1是由N的i点和参考点v_0合并而成的，在N图中点i和点v_0之间的边，在N_1中成为自环。所以对于N_1中的一个树，在N中不仅不能构成回路，而且还不形成一条从i到v_0的路径，否则，在N_1中就构成回路而不构成树。因此N_1中的树，在N中是这样一个子图，它有n_v-2条边不含回路，不连通，并使i和v_0处在二个不同的部分内。这样的子图在N中称为2-树，它的符号用2-树$(i,\ v_0)$或$2-T(i,\ v_0)$，这里i和v_0之间的“,”号，表示i和v_0处在二个部分之中。于是

$$\Delta_{ii}=\sum_{N\text{的全部2-树}(i,v_0)}\text{2-树}(i,v_0)\text{的导纳积}$$

$$=\sum_{\text{全部2-}T(i,v_0)}\text{2-}T(i,\ v_0)\text{的导纳积} \tag{8-17}$$

由此就得下面两个定理。

定理8—2：一个包含无源元件的电网络N的节点导纳行列式的对称余子式Δ_{ii}等于该网络的全部2-树（i，v_0）导纳积之和。

不对称余子式也可以用网络N直接得出。N_1的树导纳积等于网络N的2-T（i，v_0），而N_2的树导纳积等于网络N的2-T（j，v_0）。所以N_1和N_2的公树导纳积应等于N的2-T（ij，v_0）并带有一个符号$\in$，后者可以证明$\in=(-1)^{i+j}$。因此

$$\Delta_{ij}=(-1)^{i+j}\sum_{\text{全部2-}T(ij,v_0)}\in\text{2-}T(ij,v_0)\text{导纳积}$$

$$=\sum_{\text{全部2-}T(ij,v_0)}\text{2-}T(ij,v_0)\text{导纳积} \tag{8-18}$$

关于$\in$的证明过程十分冗长，有兴趣的读者可参见有关书籍。

定理8—3：一个无源网络的节点导纳矩阵行列式的不对称余子式Δ_{ij}等于该网络的全部2-T（ij，v_0）导纳积之和。

例8—2　试用拓扑公式求图8—7的不对称余子式Δ_{12}。

因为　　$\Delta_{12}=\sum\text{2-}T(12,\ 0)$ 导纳积

所以所有可能的2-T(12，0)包含2-T(123，0)和2-T(12，03)，其中

$$\text{2-}T\ (123,\ 0)=y_cy_b+y_by_d+y_dy_c$$

而　　$\text{2-}T\ (12,\ 03)=0$

所以　　$\Delta_{12}=\sum\text{2-}T(12,0)=y_cy_b+y_by_d+y_dy_c$

从式（8—15），（8—17）和（8—18）等拓扑公式还可以直接由N计算Y_{ij}^{sc}及网络函数。

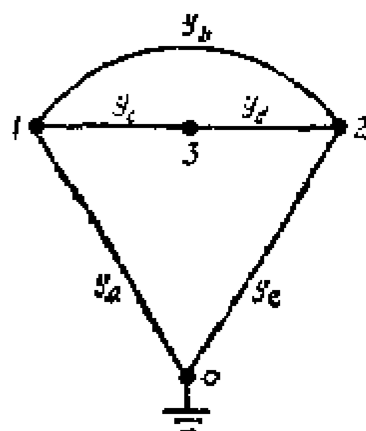


图　8—7

第六节　网络函数

网络函数定义为零状态响应的象函数与输入象函数的比，记为$H(s)$。

为了得到网络函数的拓扑公式，现在再介绍一些符号：

$$\nabla=\sum\text{树导纳积}$$

$$W_{a_1a_2a_3\cdots a_r,b_1b_2\cdots b_s}=\sum\text{2-树}\ (a_1a_2\cdots a_r,\ b_1b_2\cdots b_s)\ \text{导纳积}$$

设有图8—8所示的三端网络N，N是一个由无源元件构成的网络，图中所示的电压电流均为象函数。

根据节点方程式（8—13），有

$$\begin{pmatrix}v_p\\v_q\end{pmatrix}=\frac{1}{\Delta_n}\begin{pmatrix}\Delta_{pp}&\Delta_{qp}\\\Delta_{pq}&\Delta_{qq}\end{pmatrix}\begin{pmatrix}I_p\\I_q\end{pmatrix}$$

q 端开路时的网络函数

$$\frac{I_p}{V_p}\Big|_{I_q=0}=\frac{\Delta}{\Delta_{pp}}=\frac{\Delta}{W_{(p,0)}} \tag{8—19}$$

$$\frac{I_p}{V_q}\Big|_{I_q=0}=\frac{\nabla}{W_{(pq,0)}} \tag{8—20}$$

$$\frac{V_q}{V_p}\Big|_{I_q=0}=\frac{\Delta_{pq}}{\Delta_{pp}}=\frac{W_{(pq,0)}}{W_{(p,0)}} \tag{8—21}$$

q 端短路时的网络函数，以电流比为例，

$$\frac{I_p}{I_q}\Big|_{V_q=0}=\frac{-\Delta_{qq}}{\Delta_{pq}}=\frac{-W(q,0)}{W(pq,0)} \tag{8—22}$$

式（8—19），（8—20），和（8—21），称为开路网络函数的拓扑公式；式（8—22）为短路网络函数拓扑公式的一例。

例 8—3　试计算图 8—9 的开路网络函数$\frac{I_1}{V_1}$，$\frac{I_2}{V_2}$。

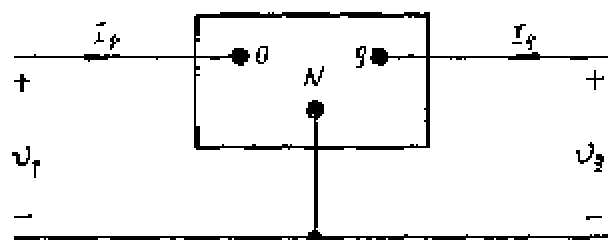


图　8—8

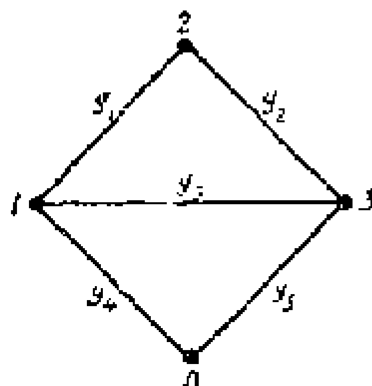


图　8—9

$$\nabla=y_1y_2y_4+y_1y_2y_5+y\ y_3y_4+y_1y_3y_5+y_1y_3y_4$$
$$+y_2y_3y_4+y_2y_3y_5+y_2y_4y_5$$
$$W(1,0)=y_1y_2+y_1y_3+y_1y_5+y_2y_3+y_2y_5$$
$$W(2,0)=y_1y_2+y_2y_3+y_3y_{\ }+y_3y_4+y_4y_5+y_3y_5+y_1y_5+y_2y_4$$

输入导纳是网络函数的特例，将上面 3 式代入下面式子，即得图 8—9 的输入导纳

$$\frac{I_1}{V_1}=\frac{\nabla}{W(1,\ 0)}$$
$$\frac{I_2}{V_2}=\frac{\nabla}{W(2,\ 0)}$$

第七节　电流图和电压图

回顾本章第三节，在导出节点分析法的过程中，利用了KCL和KVL，其中 $A_I I=0$ 中的A_I是表示电流拓扑关系的缩减关联矩阵，而$V=A_V^t V_n$中的A_V^t是电压拓扑关系中的转置缩减关联矩阵。由此看出$Y_n=A_I Y A_V^t$中A_I和A_V^t本来是分别指电流和电压的拓扑关系的。但在无源元件的网络中电流和电压的拓扑关系相同，所以是用同一个图表示的。但是在含有有源元件的网络中情况就不同了。为此需要引出电流图和电压图的概念。

一个导纳 y 可以定义为：

$$yv_{r,s}=i_{t,u} \qquad (8-23)$$

式中　$v_{r,s}$——指由点 r 到点 s 的电压；

　　　$i_{t,u}$——指由点 t 到点 u 的电流。

这个导纳 y 可用图 8—10表示，图中$v_{r,s}$ 代表 y 的电压边，$i_{t,u}$代表 y 的电流边，所以在▽箭头中分别标以 v 和 i 以示区别。

一个三端跨导，根据定义，即式（8—23）应为

$$gv_{h,k}=i_{p,k} \qquad (8-24)$$

就是说电压边和电流边的终端是一个点，如图 8—11所示。

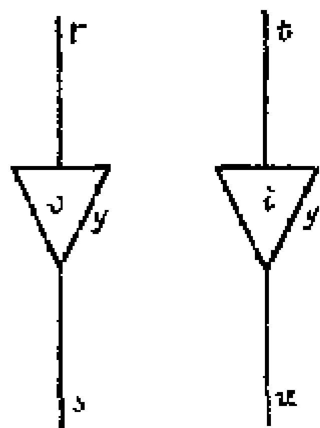


图　8—10

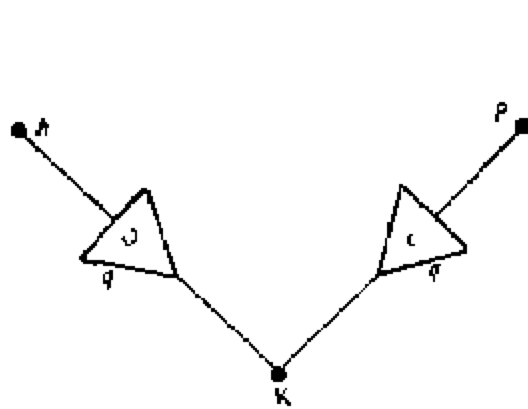


图　8—11

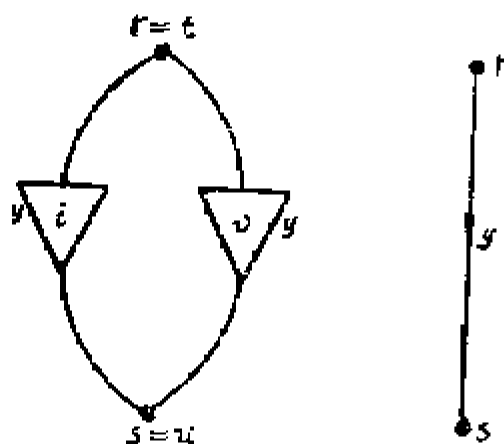


图　8—12

假如式（8—23）中 $r=t$，$s=u$，则导纳可以表示为图 8—12。这种情况，y 就是一个无源导纳，即

$$yv_{r,s}=i_{r,s} \qquad (8-25)$$

因此就可以用一条权为 y 的边来代表，一般不区别电压边和电流边。

假如式（8—23）中 $r=u$，$s=t$，则得

$$yv_{r,s}=i_{s,r}$$

因为
$$i_{s,r}=-i_{r,s}$$

所以
$$(-y)v_{r,s}=i_{r,s} \qquad (8-26)$$

式（8—26）表示的是一个负导纳。由以上分析，可以清楚地得到下面二个结论：

（一）无源元件的定向是任意的，面有源元件则不能；

（二）无源元件一般常用一条边权为 y 的边来表示，这只是一种简化，实际上元件 y 都应当用二条边表示，即电压边和电流边。

为了熟悉电压边和电流边，下面研究一下式（8—12）。对于一个 4 点的网络 N 来说，式（8—12）为

$$\begin{pmatrix} y_{11} & y_{12} & y_{13} \\ y_{21} & y_{22} & y_{23} \\ y_{31} & y_{32} & y_{33} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} i_1 \\ i_2 \\ i_3 \end{pmatrix}$$

式中　v_p—— p 点到参考点的电压（$p=1$，2，3）；

　　　i_p——参考点到 p 点的电流（$p=1$，2、3）。

因此对应于网络 N 的线性图为图 8—13。

图 8—13是一种很有用的表示方法。当导纳 y 的电压边和电流边是任意的，就是说它满足式（8—23），那么，y 在节点导纳矩阵 Y 中的位置就值得注意了。因为

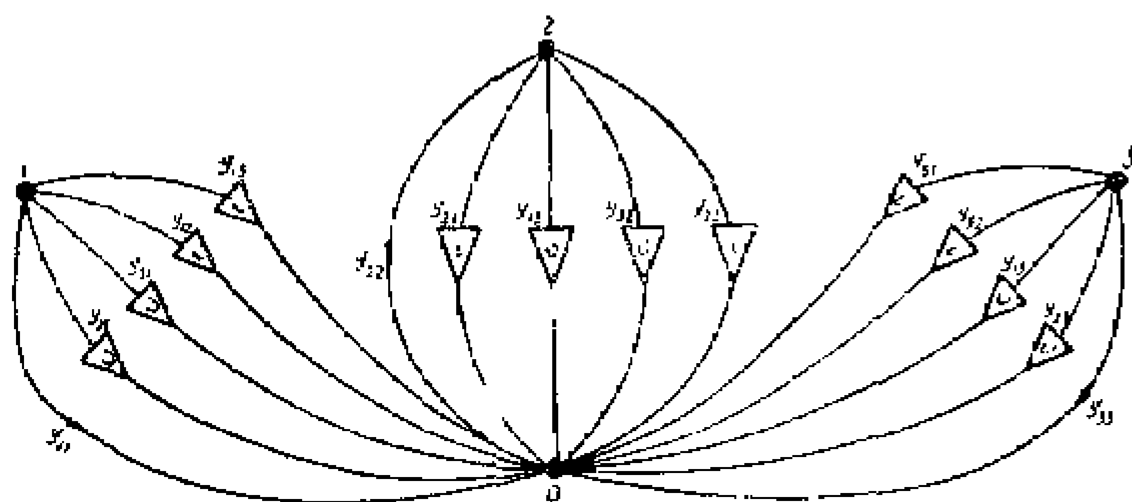


图 8—13

$$yv_{r,s}=i_{t,u}$$
$$y(v_r-v_s)=i_{t,u} \qquad (8-27)$$

式中，v_s和v_r分别是r，s点的节点电压。

设网络N含有上述电导y，它的节点方程为

$$\begin{pmatrix} y_{11} & y_{12}\cdots y_{1n} \\ y_{21} & y_{22}\cdots y_{2n} \\ \cdots & \cdots \\ y_{n1} & y_{n2}\cdots y_{nn} \end{pmatrix}\begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}=\begin{pmatrix} i_1 \\ i_2 \\ \vdots \\ i_n \end{pmatrix}$$

节点导纳矩阵的第t行乘电压列得

$y_{t1}v_1+y_{t2}v_2+\cdots+y_{tn}v_n=i_t=$节点$t$的电流源和。从式（8—27）可以看出，$y$一定位在$Y_n$中（$t$，$r$）元素，而$-y$位在（$t$，$s$）元素。同样$-y$位在（$u$，$r$）元素，而$y$位在（$u$，$s$）元素上。

下面我们来看看例8—1。从例中Y_n可见，Y_n能分成几个矩阵的和。在没有M的情况下，网络由g_a，g_d L_b，L_c和C_e五个无源元件组成的节点导纳矩阵Y_n为：

$$\begin{pmatrix} g_a+\frac{1}{SL_b} & -\frac{1}{SL_b} & 0 \\ -\frac{1}{SL_b} & \frac{1}{SL_b}+\frac{1}{SL_c}+g_d & -\frac{1}{SL_c} \\ 0 & -\frac{1}{SL_c} & \frac{1}{SL_c}+SC_e \end{pmatrix}=\begin{pmatrix} g_a & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}+\begin{pmatrix} \frac{1}{SL_b} & -\frac{1}{SL_b} & 0 \\ -\frac{1}{SL_b} & \frac{1}{SL_b} & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$+\begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{SL_c} & -\frac{1}{SL_c} \\ 0 & -\frac{1}{SL_c} & \frac{1}{SL_c} \end{pmatrix}+\begin{pmatrix} 0 & 0 & 0 \\ 0 & g_d & 0 \\ 0 & 0 & 0 \end{pmatrix}+\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & SC_e \end{pmatrix}$$

如果有互感存在，则还需增加二个子矩阵

$$\begin{pmatrix} 0 & -\frac{M}{S\Delta} & \frac{M}{S\Delta} \\ 0 & \frac{M}{S\Delta} & -\frac{M}{S\Delta} \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ -\frac{M}{S\Delta} & \frac{M}{S\Delta} & 0 \\ \frac{M}{S\Delta} & -\frac{M}{S\Delta} & 0 \end{pmatrix}$$

由此可见，无源元件在 Y_n 中的位置处于对称主对角线位置，若在例 8 －1 中增加一个受控电流源，如图 8 —14所示。

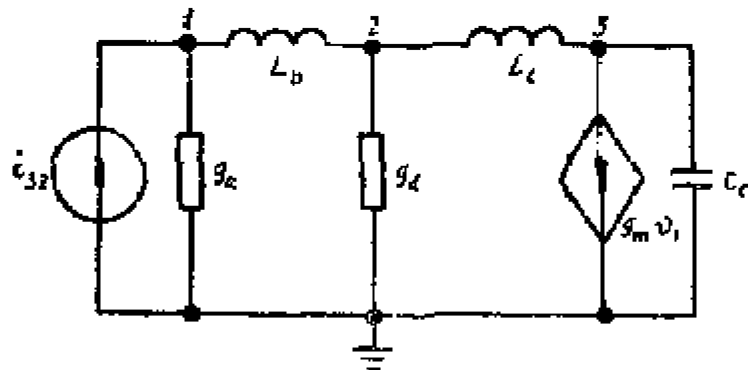


图 8 —14

那么，在节点导纳矩阵的 g_{31} 元素上应当是 g_m。根据上面的例子，对几种有源元件的拓扑表示法分析如下：

（一）互感线圈

互感线圈的原理图和等值电路图表示在图 8 —15（a）和（b）上。

图 8 —15的电感矩阵为：

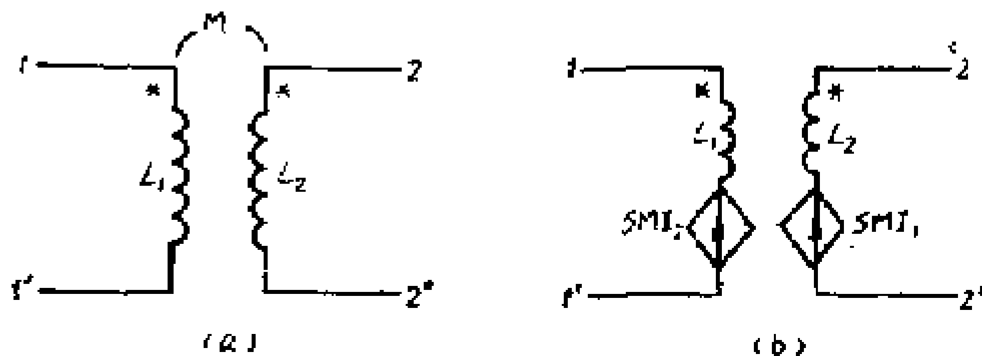


图 8 —15

$$L = \begin{pmatrix} \frac{L_2}{S\Delta} & \frac{-M}{S\Delta} \\ \frac{-M}{S\Delta} & \frac{L_1}{S\Delta} \end{pmatrix}$$

式中

$$\Delta = \begin{pmatrix} L_1 & M \\ M & L_2 \end{pmatrix}$$

这样，这个互感线圈的拓扑表示如图 8 —16所示

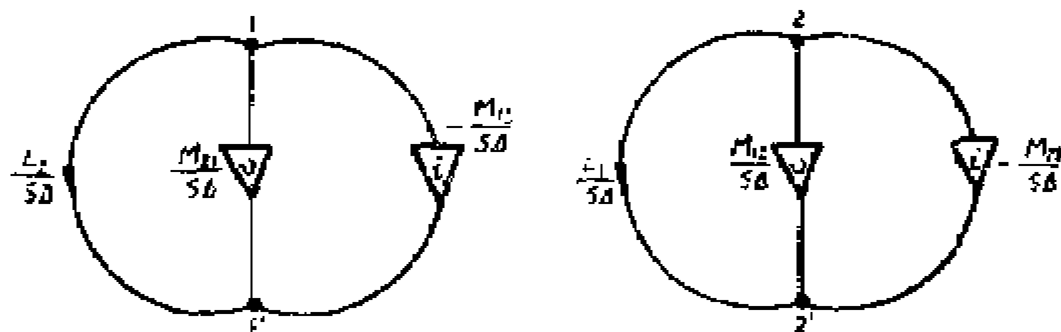
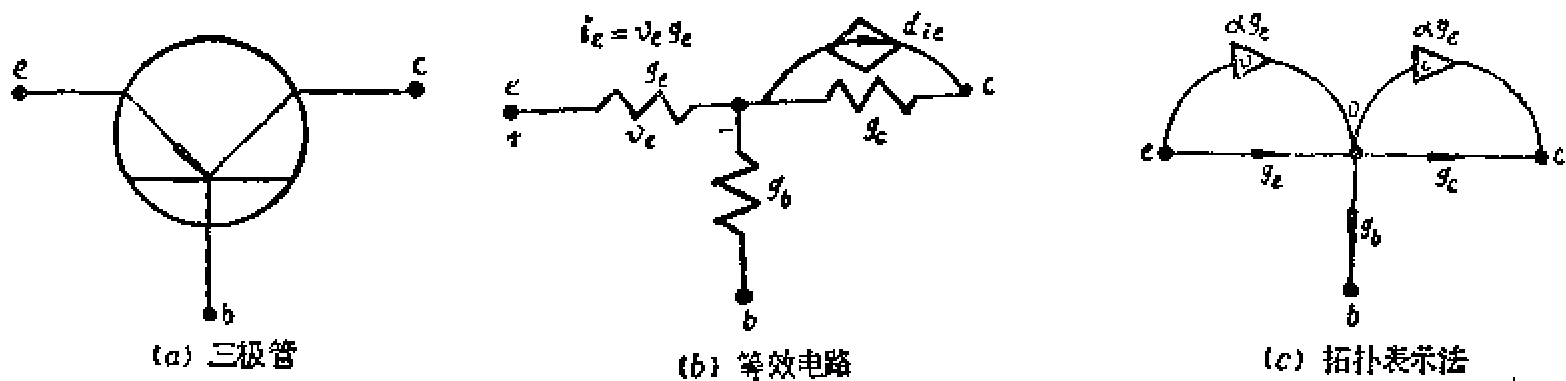


图 8 —16

（二）晶体三极管

晶体三极管原理图、等效电路和拓扑表示如图 8 —17中（a），（b）和（c）所示。网络的线性图可以由元件线性图构成。



(a) 三极管　　(b) 等效电路　　(c) 拓扑表示法

图 8—17

例 8 — 4　图 8 —18为一网络图，它的线性图为图 8 —19。

图中
$$y_1=\frac{L_2}{S\Delta}\qquad y_2=y_3=\frac{-M_{12}}{S\Delta}$$
$$y_4=\frac{L_1}{S\Delta}$$

上述线性图是电流，电压都在一个图上，称为混合图。一个混合图可以分成二个图，其中一个只含电流边和无源元件，但包含混合图的全部点称为电流图；另一个图只含电压边和无源元件，也含全部节点，称为电压图；分别由图 8 —20的（a）和（b）所示。
显然，当网络只含无源元件时，电流图和电压图是同样的。

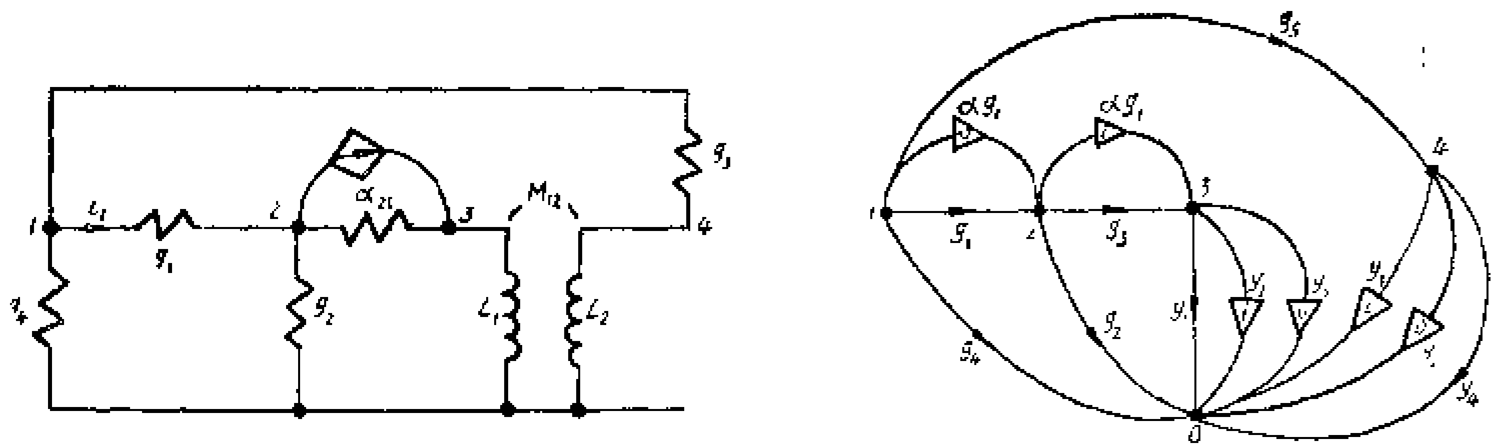
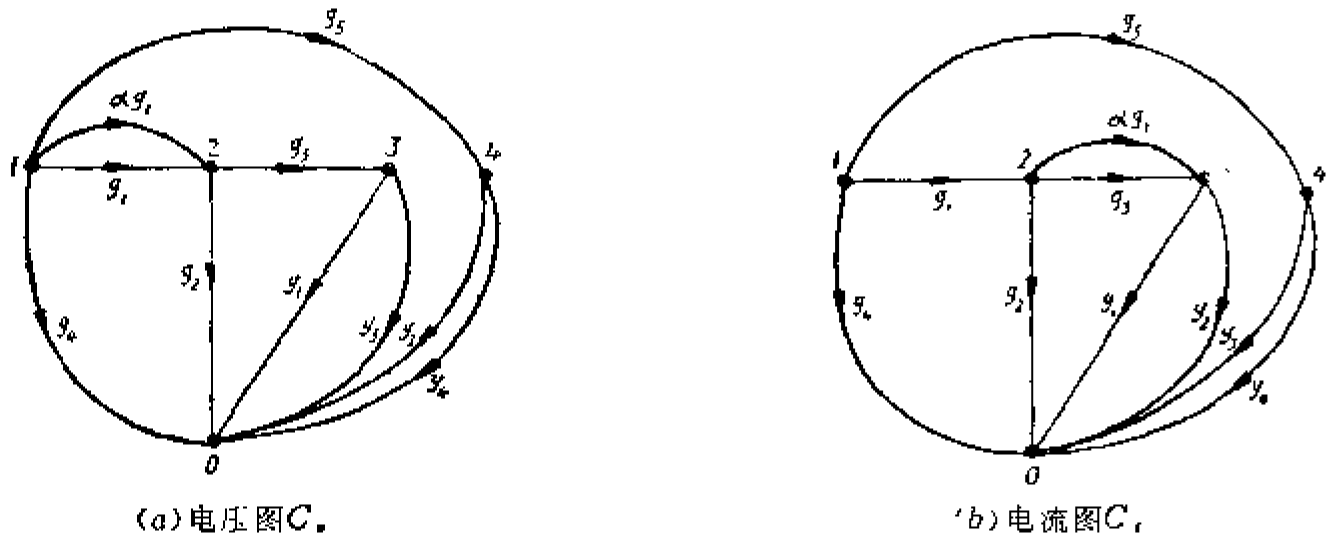


图 8 —18　　　　图 8 —19



(a)电压图C_v　　(b)电流图C_i

图 8 —20

第八节　有源网络的拓扑分析

有源网络的拓扑分析步骤如下：

(一) 将图中元件的电原理图转换成等值电路，无源元件不变，有源元件按表 8 — 1 转换：

电原理图与等值电路的转换　　　　表 8 — 1

	电原理图	等效电路
半导体三极管		
放大器		
运算放大器		
电子管		

(二) 将等值电路图转换成混合图，再作出电压图和电流图。

(三) 根据电流图和电压图分别写出关联矩阵。

则

$$Y_n = A_{ii} Y A_v^t$$

根据与无源网络类似的方法可以推出

$$\det Y_n = \sum_{全部公树} \epsilon_k G_i 和 G_v 的公树导纳积$$

式中　$\epsilon_k = \pm 1$

为了确定 ϵ 的正负号，下面介绍主点概念。设 G_i 是有向图的一棵生成树．生成树的树支数等于 $n_v - 1$。主点是对某一条树枝而言的。假设边 y 的两个端点为 u，v，由于树不含回路，所以从 u 到参考点，和从 v 到参考点的两条路径中，只有一条经过 y 边。于是可以定义：若 y 的某一端点 u 到参考点的路径中包含 y 边，则 u 称为 y 的主点。

例 8 — 5　图 8 —21是一棵树。

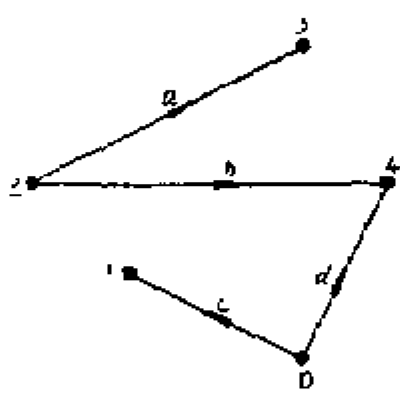


图　8 —21

边和主点

表 8 — 2

边	主　点
a	3
b	2
c	1
d	4

设以 0 为参考点，则各边的主点示于表 8 — 2 。

由此可见：参考点不作为任一边的主点；n_v-1个点正好作 n_v-1条树枝的主点；每个点只作一条边的主点。

假设 G_v 和 G_i 是两个图，每个图对应着一棵生成树，并且两棵树包括相同的导纳，相同的

点，如图 8 —22所示。那么，对于每个图每条边的主点，表示于表 8 — 3 中。

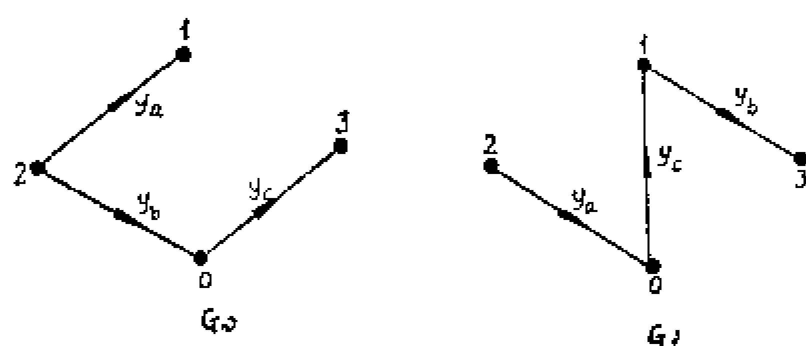


图 8 —22

边和主点

表 8 — 3

边	主点	
	G_v	G_i
y_a	1	2
y_b	2	3
y_c	3	1

对于两棵有相同导纳的树，其正负号置换，记SP，有如下规则：

（一）G_v和G_i各自排列成一行，而以主点为列，构成一个二行（n_v-1）列的表格记为SP；

（二）表中某一元素是对应行和列的导纳；

（三）如果G_v行，以 1 为主点的导纳y_a是指向1的，则写成y_a^-，离开主点的导纳为y_a^+；G_i类同。

所以图 8 —22的SP为：

$$SP=\begin{matrix} \\ G_v \\ G_i \end{matrix}\begin{pmatrix} 1 & 2 & 3 \\ y_a^- & y_b & y_c^- \\ y_c^- & y_a & y_b^- \end{pmatrix}$$

把第二行元素重新排列，使换成与第一行相同顺序。如果置换次数为偶数，则称偶置换；如置换次数为奇数，则称奇置换。根据置换的奇偶性可以确定正负号置换值，记为$\nabla[SP]$。奇置换的$\nabla[SP]=-1$，而偶置换的$\nabla[SP]=1$。所以图 8 —22的

$$\nabla[SP]=\nabla\left[\begin{pmatrix} y_a^- & y_b & y_c^- \\ y_c^- & y_a & y_b \end{pmatrix}\right]=1$$

根据正负号置换值可以确定$\in_k$。关于它的证明，由于十分冗长，这里就从略了。

定理 8 — 4：设 $(y_{k_1}, y_{k_2}, \cdots, y_{k_{n_v-1}})$是一棵公树中的诸边，其中 $T_i=(y_{k_1}, y_{k_2}, \cdots, y_{k_{n_v-1}})$ 是G_i中的一棵生成树，而$T_v=(y_{k_1}, y_{k_2}, \cdots, y_{k_{n_v-1}})$是$G_v$中的一棵生成树。相应地，$A_{fi}(k_1, k_2, \cdots, k_{n_v-1})$ 是T_i的缩减关联矩阵，而$A_{fv}(k_1,k_2,\cdots, k_{n_v-1})$ 是T_v的缩减关联矩阵。G_i和G_v有相同的参考点。那么

$$\begin{aligned}\in_k&=|A_{fi}(k_1,k_2,\cdots,k_{n_v-1})|\cdot|A_{fv}(k_1,k_2,\cdots,k_{n_v-1})|\\&=\nabla[T_i\text{和}T_v\text{的}SP]\end{aligned} \qquad (8—28)$$

因为可以直接从T_i和T_v得到每条边的主点，所以我们就可以直接从 T_i和T_v的正负号置换中得到$\in_k$。

很明显，如果一棵公树只包括些无源元件，那么 $A_{fi}(k_1, k_2,\cdots,k_{n_v-1})$和 $A_{fv}(k_1, k_2, \cdots, k_{n_v-1})$是相同的，因此得到下列的规定：

如果一棵公树只包括些无源元件，公树的$\in_k$是＋ 1 。

习　题

8 — 1　利用电源转移，画出图 8 —23中网络的图。

8－2　用拓扑公式求图 8－24的节点导纳行列式之值

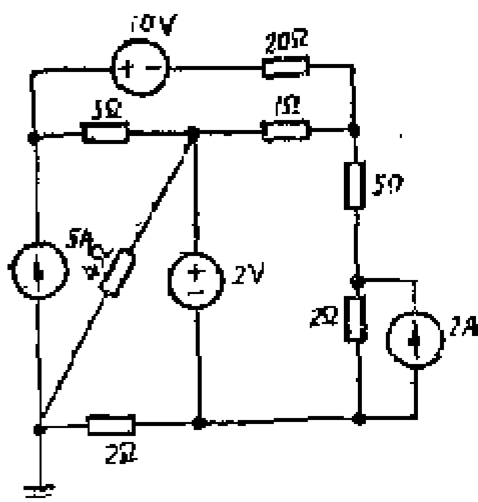


图　8－23

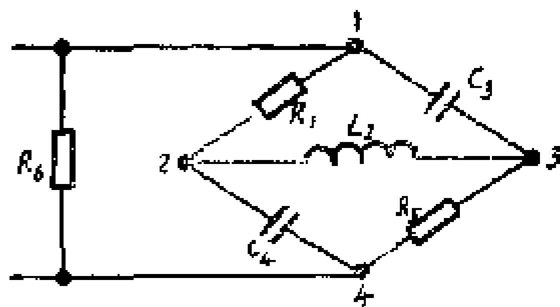


图　8－24

8－3　试写出RLC网络的回路阻抗矩阵Z_l的(i, j)元素的余子式。

8－4　列出图8－25的所有生成树和所有生成 2-树（31,5）。

8－5　计算图 8－26的转移网络函数V_{25}/V_{10}

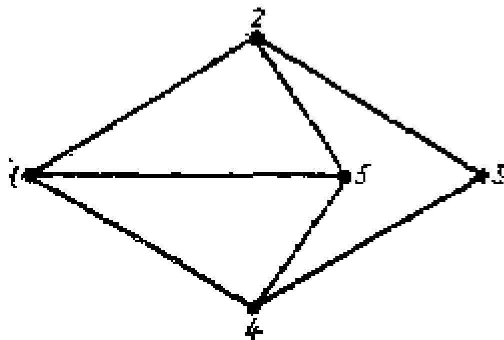


图　8－25

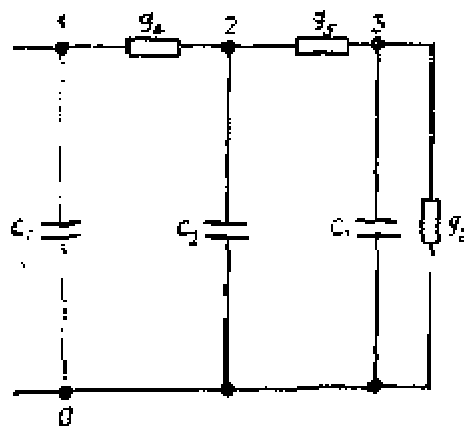


图　8－26

8－6　将图 8－26的网络看成为一个四端网络，输入端为1-0，输出端为2-3。求参数y_{11}，y_{12}，y_{21}，和y_{22}。

8－7　画出图 8－25的对偶网络，然后研究两个网络的各个量之间的关系，如象一个网络中的回路方程是另一个网络的节点方程。

8－8　按步骤一步一步地导出回路矩阵方程

$$B_f Z(s) B_f^t I_s(s) = E(s)。$$

8－9　二端RLC网络如图 8－27所示。证明它的驱动点（1，r）导纳为

$$\frac{I_1(s)}{V_2(s)} = \frac{\sum 树导纳积}{\sum 2\text{-树}(1,r)导纳积}$$

8－10　一个四端网络有 4 个短路导纳公式，导出 RLC 两端网络的各个拓扑公式。网络有一个公共参考点 r，如图 8－28所示。

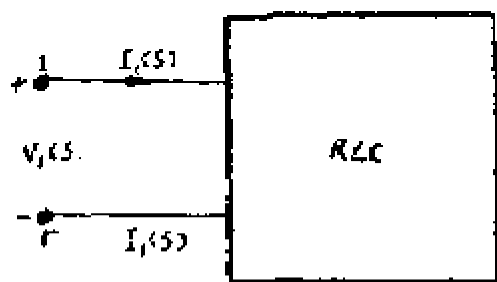


图　8－27

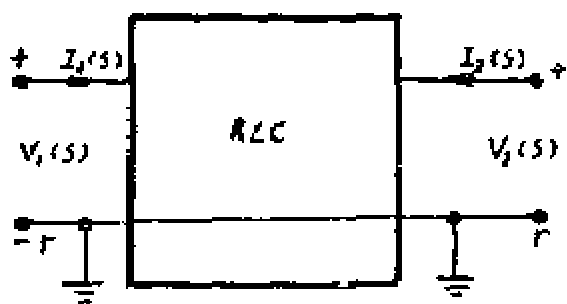


图　8－28

第九章　流图和信号流图

第一节　流　　图

流图是一种加权有向图，它满足下列条件：

（一）流图G_f包括n个点；

（二）G_f中任意一个点i到点j的一条有向边都有边权w_{ij}。

图9－1就是一个$n=3$的流图。

流图的边权构成一个方阵W，对图9－1的流图来说，它的对应矩阵为

$$W=\begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{pmatrix} \qquad (9-1)$$

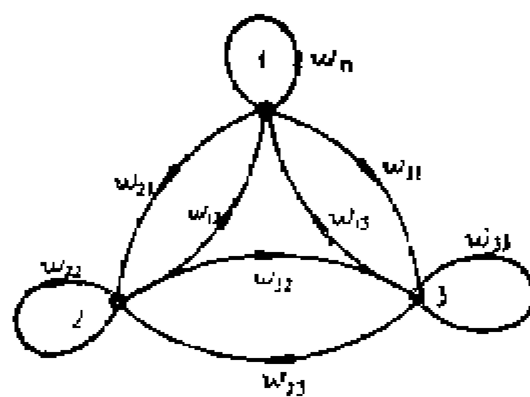


图　9－1

这就是说一个方阵对应一个流图。

从流图可以得到矩阵的行列式，根据行列式性质

$$\det W=\sum_{j_1j_2\cdots j_n}(-1)^{\tau(j_1j_2\cdots j_n)}w_{1j_1}w_{2j_2}\cdots w_{nj_n}$$

$$=\sum_{j_1\cdots j_n}\delta(j_1j_2\cdots j_n)w_{1j_1}w_{2j_2}\cdots w_{nj_n} \qquad (9-2)$$

如果（一）$j_1j_2\cdots j_n$不是1，2，…，n的一个排列，则$\delta(j_1j_2\cdots j_n)=0$

（二）$j_1j_2\cdots j_n$是1，2，…，n的一个偶排列，则$\delta(j_1j_2\cdots j_n)=1$

（三）$j_1j_2\cdots j_n$是1，2，…，n的一个奇排列，则$\delta(j_1j_2\cdots j_n)=-1$

因为流图G_f与矩阵W相对应，所以$\det W$中的每一项对应着G_f的一个子图g，这个子图包含G_f的n条边，n个点，它是一个有向回路或若干个有向回路的点不共用并集。因为在这种情况下，$w_{1j_1}w_{2j_2}\cdots w_{nj_n}$的下标中，每个点$i$都出现两次。如果g是连接的，它就是一个有向回路；如果g有k个部分，那么它就是k个有向回路的点不共用并集。

为了方便起见，下面定义一个概念，叫P-组循环，它的定义是：

一个P-组循环是有向图G的一个子图，它包括G的全部点，它是一个有向回路或若干个有向回路的点不共用并集，记为D_j。

所以式（9－2）可以写成

$$\det W=\sum\varepsilon_j P\text{-组循环乘积}=\sum\varepsilon_j D_j \qquad (9-3)$$

式中ε_j是由P-组循环D_j的排列性质确定的，也可以由W的阶数n和这个循环的部分数k_j来确定。可以证明

$$\det W=(-1)^{n}\sum_{j}(-1)^{k_j}P\text{-组循环乘积} \qquad (9-4)$$

式中k_j就是第j个P-组循环的部分数。

例9—1　图9—1的矩阵W如式（9—1）所示。

$$\det W=(-1)^3[(-1)^3w_{11}w_{22}w_{33}+(-1)^2w_{12}w_{21}w_{33}+(-1)^2w_{13}w_{22}w_{31}$$
$$+(-1)^2w_{11}w_{23}w_{32}+(-1)w_{12}w_{23}w_{31}+(-1)w_{13}w_{21}w_{32}]$$

每个P-组循环对应的子图为：

图中：（a）是　$D_1=w_{11}w_{22}w_{33}$

（b）是　$D_2=w_{12}w_{21}w_{33}$

（c）是　$D_3=w_{13}w_{22}w_{31}$

（d）是　$D_4=w_{11}w_{23}w_{32}$

（e）是　$D_5=w_{12}w_{23}w_{31}$

（f）是　$D_6=w_{13}w_{21}w_{32}$

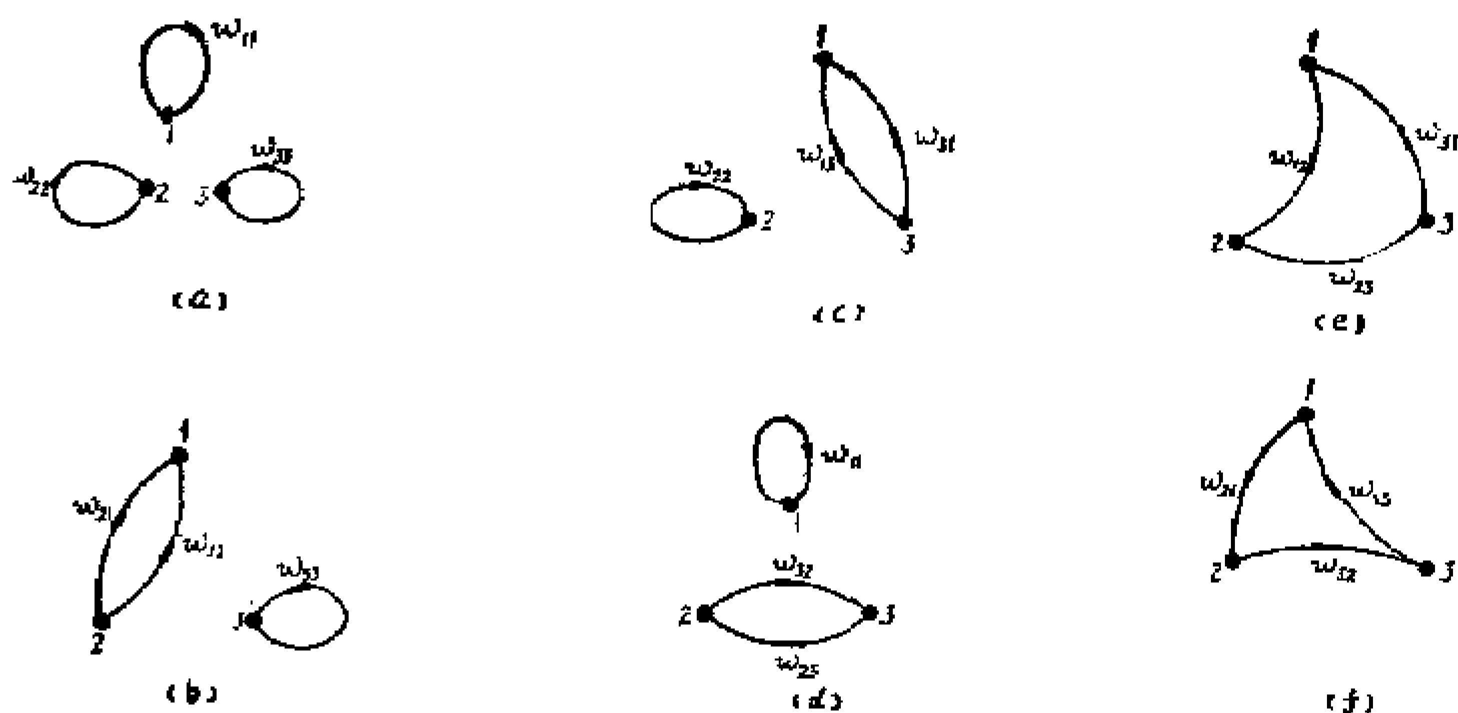


图　9—2

第二节　信号流图

信号流图G_s也是一个加权有向图，它除了给各边赋有边权外，并给各点赋有点权。点权称为节点信号，边权称为支路传输或支路增益。

图9—3是一个加权有向图，图G_s含有$n+1$个点，每个点的点权为$x_0x_1\cdots x_n$，点j到点i的边权为m_{ij}，所以G_s表示一个信号流图。

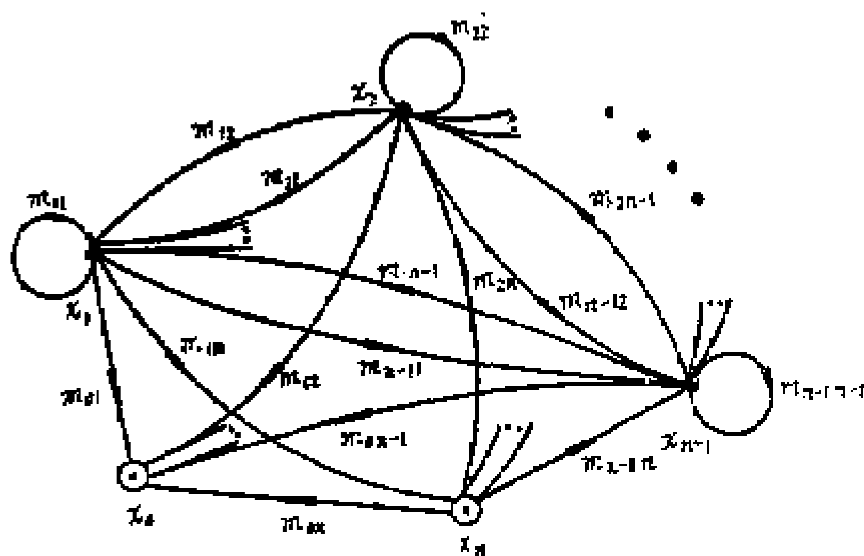


图　9—3

从信号流图可以看出，信号流图中的点可以分为三类：

第一类：它只有外向的支路，而没有内向支路。例如点x_0这类节点，表示信号只有向外

传输，所以称为源点。

第二类：它只有内向支路，而没有外向支路。例如x_0这类节点，表示信号只有向着x_0传输而没有从x_0传出，所以称为收点。

第三类：称为一般节点，它有内向支路，也有外向支路，所以信号可以通过它传输给别的节点。

信号流图中的节点i的信号传输到节点j时，用（i指向j）有向支路传输m_{ji}表示，如图9—4所示。

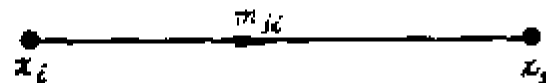


图 9—4

节点信号x_i通过支路传输m_{ji}传输到x_j时，信号增长m_{ji}倍，即：

$$x_j = m_{ji}x_i \tag{9—5}$$

有向支路还表示了由于x_i的原因，产生了x_j这个结果的一个因果关系。

如果一个节点信号是由几个节点信号传输而来的，则此节点信号就等于流入该点信号之总和。如图9—5中，x_i节点信号是由k个节点传输而来的，则x_i就等于各节点信号和支路传输乘积之和。记为：

$$x_i = \sum_k m_{ij}x_j \tag{9—6}$$

应当指出，图9—5中点x_{k+1}的信号是不能传输给x_i的。因为x_i表现为$x_1x_2\cdots x_k$的果，而它的因只是$x_1x_2\cdots x_k$，不包括x_{k+1}。

根据上述约定，信号流图9—3的各节点信号表示为下面方程组：

$$\left.\begin{aligned}
x_1 &= m_{11}x_1 + m_{12}x_2 + \cdots + m_{1n}x_n \\
x_2 &= m_{21}x_1 + m_{22}x_2 + \cdots + m_{2n}x_n \\
&\vdots \\
x_{n-1} &= m_{n-1\,1}x_1 + m_{n-1\,2}x_2 + \cdots + m_{n-1\,n}x_n \\
x_0 &= m_{01}x_1 + m_{02}x_2 + \cdots + m_{0n}x_n
\end{aligned}\right\} \tag{9—7}$$

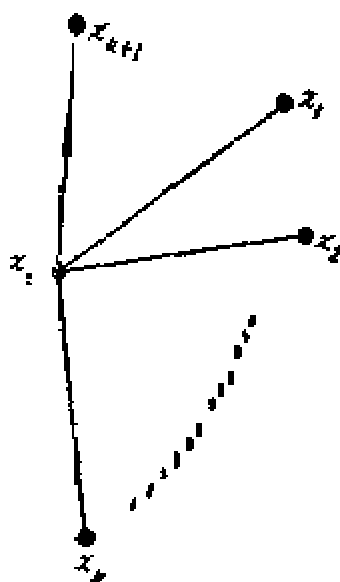


图 9—5

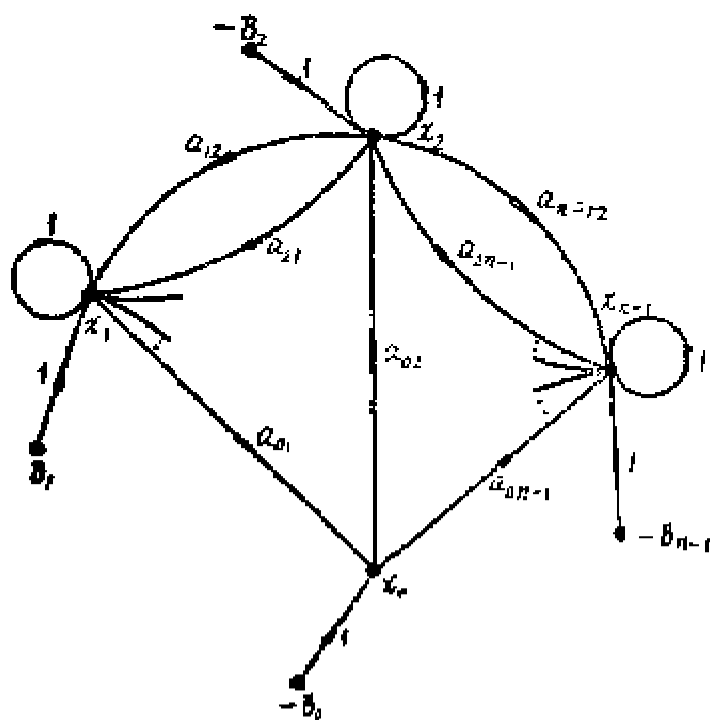


图 9—6

从上式可见，源点x_n不出现在方程式左方，而收点x_0不出现在方程的右方。

我们还看到了，一组线性代数方程组和一个信号流图相对应，这就可以利用信号流图来分析求解线性方程组。换句话说，若我们所要分析的物理系统可以最后归纳为解决一组线性

代数方程组的时候，这个物理系统就可以用信号流图来分析。应当指出，用信号流图解线性代数方程常常比矩阵法要快些，更主要的它比矩阵法更显示出变量之间的因果关系。为此，我们首先研究一线性代数方程和信号流图之间的对应关系。

前面已经介绍，信号流图 9 — 3 和线性方程组（9 — 7）相对应。现在我们把方程（9 — 7）改写成代数方程的一般形式

$$AX=B \qquad (9-8)$$

式中

$$A=\begin{pmatrix} 1 & a_{12} & \cdots & a_{1\,n-1} & 0 \\ a_{21} & 1 & \cdots & a_{2\,n-1} & 0 \\ \vdots & & & & \\ a_{n-1\,1} & & \cdots & 1 & 0 \\ a_{01} & & \cdots & a_{0\,n-1} & -1 \end{pmatrix}$$

$$X=[x_1\ \ x_2\cdots x_{n-1}\ \ x_0]'$$

$$B=[B_1\ \ B_2\cdots B_0]'$$

式（9 — 8）的信号流图表示在图 9 — 6 中。

图中有 n 个点，其中 x_0是收点，此外，还有 n 个源点。这种信号流图和图 9 — 3 没有本质区别。如果将所有源点合并为一个，而适当改变源点的各支路传输，那么，图 9 — 6 就转化为图 9 — 3。

下面对一个具体电路作出它的信号流图。例 9 — 2 给出一个电子管放大电路如图 9 — 7 所示。根据图 9 — 7，列出下面方程组：

$$v_g=v_s+700i$$

$$v_1=-20v_g-700i$$

$$i=\frac{1}{9000}(v_1-v_2)$$

$$v_2=30,000i$$

$$v_3=-700i$$

在图 9 — 7 中，v_s为输入，v_2为输出，方程式中有一个变量，所以信号流图有 5 个节点和 1 个源点，如图 9 — 8 所示。

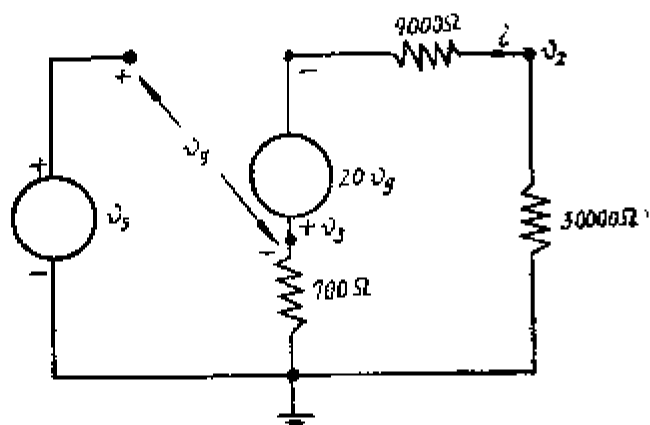


图 9 — 7

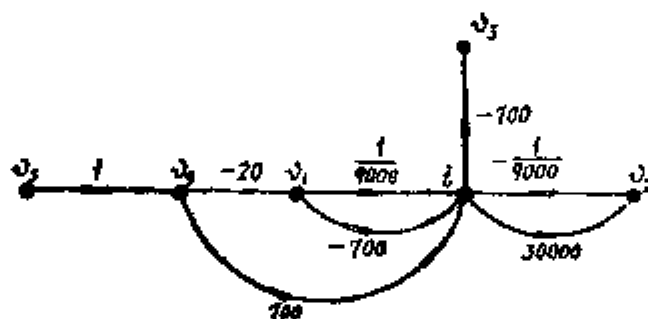


图 9 — 8

为利用信号流图来求解线性方程组，下面阐述信号流图的几个基本简化规则。

（一）支路传输的加法规则

在两点x_p和x_q之间有多条同方向支路，则它们间的支路传输等值于各支路传输之和。

图 9 — 9 中x_p和x_q间支路传输为

$$m_{qp}=\sum_{i=1}^{k}m_i \tag{9—9}$$

这是因为
$$x_q=m_1x_p+m_2x_p+\cdots+m_kx_p$$
$$=x_p\sum_{i=1}^{k}m_i=m_{qp}x_p$$

所以
$$m_{qp}=\sum_{i=1}^{k}m_i$$

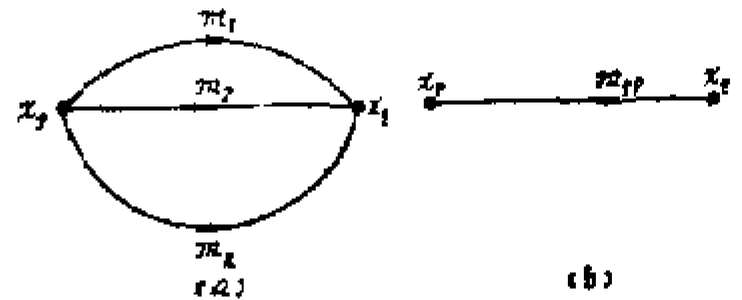


图 9—9

因此，图 9—9（a）的等值图为（b）。

（二）自环消除规则

流图中某一点x_p上有一个自环，如图 9—10（a）所示。

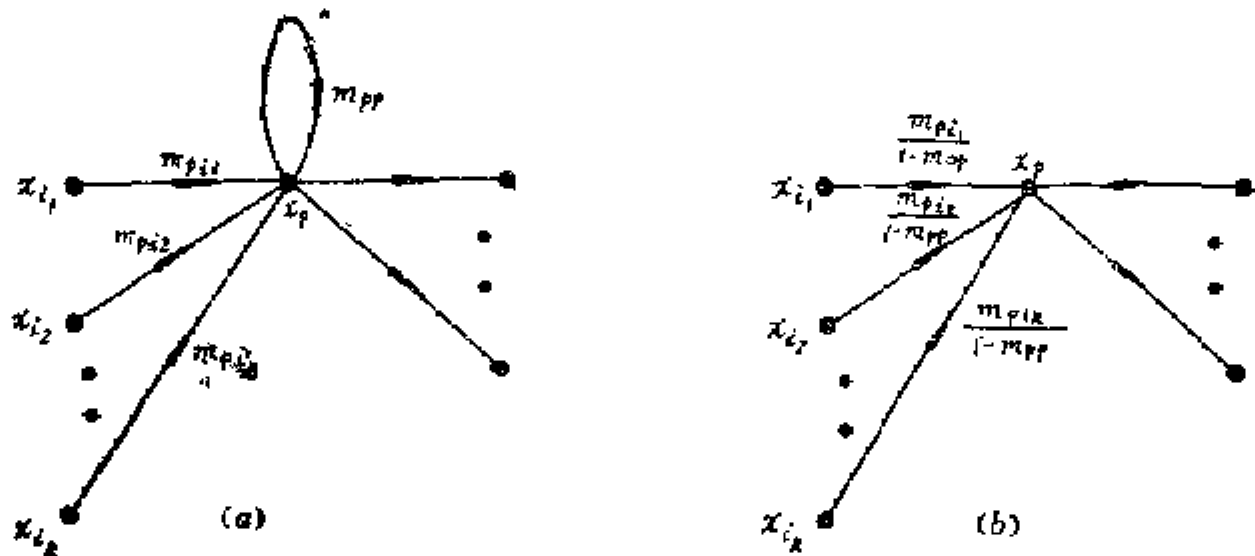


图 9—10

在x_p点的方程应为
$$x_p=m_{pp}x_p+\sum_{r=1}^{k}m_{pi_r}x_{i_r}$$

移项得
$$(1-m_{pp})x_p=\sum_{r=1}^{k}m_{pi_r}x_{i_r}$$
$$x_p=\sum_{r=1}^{k}\frac{m_{pi_r}}{1-m_{pp}}x_{i_r} \tag{9—10}$$

上式代表一个无自环的节点，它的支路数就是图（a）支路数减去自环，而支路传输为图（a）中各支路传输除以$1-m_{pp}$，于是得图（b）。

显然，自环吸收的条件是$m_{pp}\neq 1$，否则、吸收后支路传输将没有意义。

（三）串联边复合（乘法规则）

某一节点信号x_p，连续经过多个支路传输而至点x_q，如图 9—11（a）所示

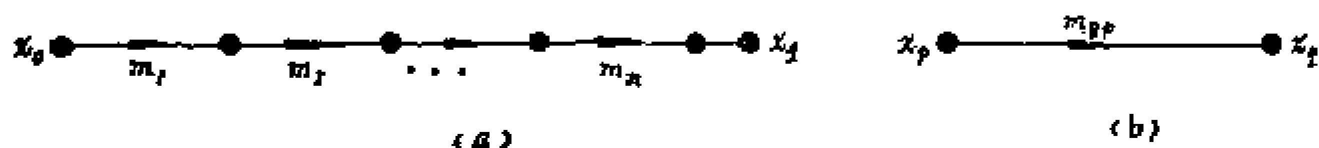


图 9—11

因为 $$x_q=m_1m_2\cdots m_nx_p=m_{qp}x_p$$

所以 $$m_{qp}=\prod_{i=1}^{n}m_i \qquad (9-11)$$

即 n 个同方向的串联支路的支路传输等值于各支路传输的积。m_{qp} 也称为 x_px_q 有向路径的传输增益。

（四）节点吸收

节点吸收的情况大致可分下面几种，它们对应的等值关系分别表示在图 9—12（a）和（b）之中。

根据上述 4 种基本规则，简化下面几个图形：

试以图 9—13中的图（a）为例，说明由左图简化为右图的过程如下。

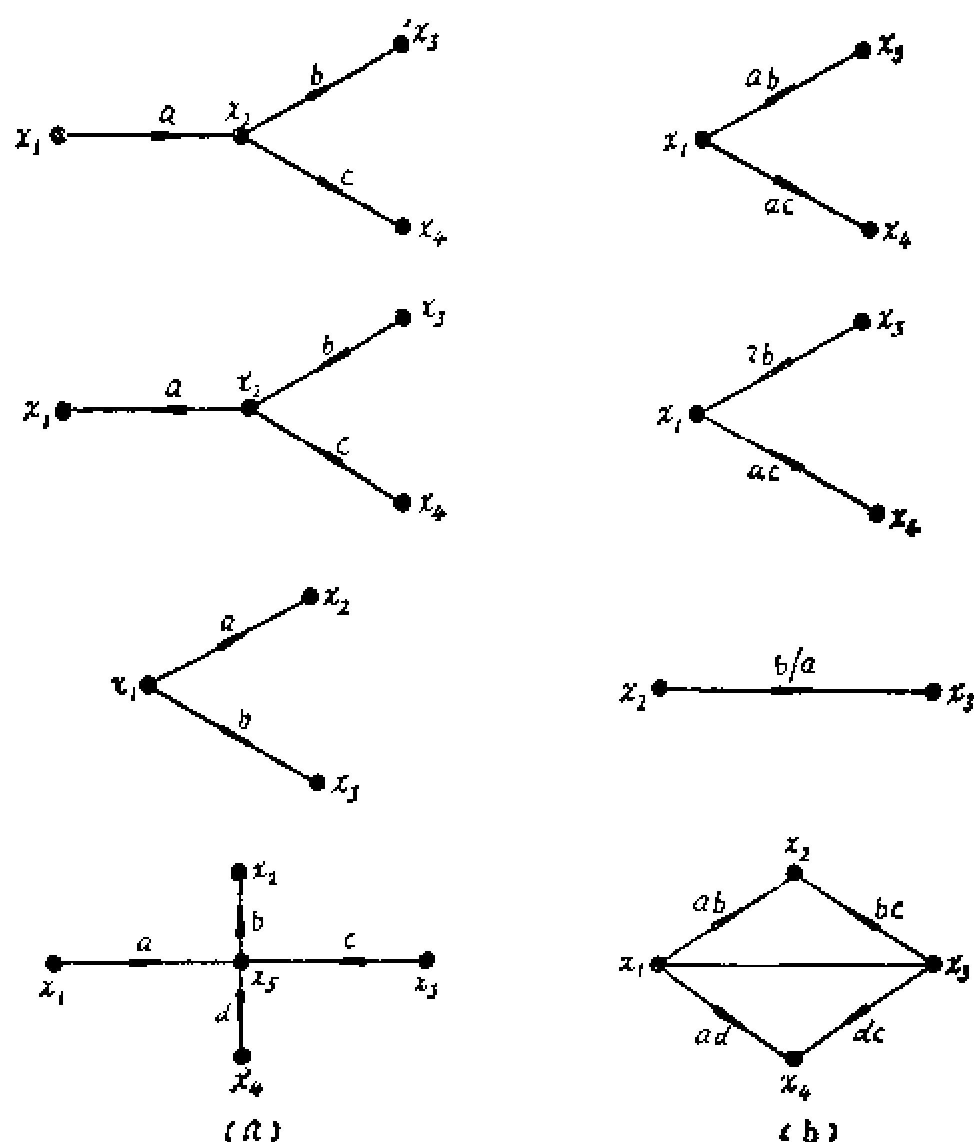


图 9—12

首先通过自环消除规则，将x_2简化为无自环的节点，得图 9—14。

x_1到x_2的支路传输是根据式（9—10）得到，即

$$x_2=\frac{1}{1-a}x_1$$

所以x_1到x_2的支路传输简化为$\frac{1}{1-a}$。

第二步是通过串联边复合，将x_2点消除。因为串联边复合后，根据式（9—11）得

$$x_3=\frac{1}{1-a}\times 1=\frac{1}{1-a}$$

这就是图 9—13(a)的右图。图 9—13中(b)、(c)、(d)、(e)、(f)的简化类推。

(a)

(b)

(c)

(d)

(e)

(f)

图 9—13

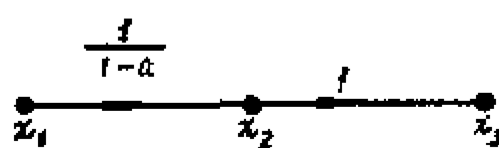


图 9—14

应用信号流图的目的是寻找输入点x_n和输出点x_0的关系。应用上述规则可以把一个复杂图形简化为一个比较简单的图。这种等效图称为剩余图。从上面的例子可以看出简化过程实际就是合并一些点和边，若连续使用上述规则，最后得到的剩余图只有x_0和x_n两点，则这两点间的权数就是输入和输出的传输值。

第三节 梅森公式

通过线性方程组得到的信号流图，逐步进行化简，可求得信号流图的传输值，这种方法，在信号流图结构比较复杂的时候，简化过程是相当麻烦的。为此，梅森提出了一种不需通过简化信号流图而直接求解传输值的方法，称为梅森增益公式或简称梅森公式。

梅森公式给出了信号流图中从源点到收点的传输值，即输出信号与输入信号的比。

$$\frac{x_0}{x_n}=\frac{\sum P_i\cdot C(\overline{P_i})}{C} \tag{9—12}$$

式中 $C=$式（9—8）中的$\det A=1-C_1+C_2-C_3\cdots$

梅森公式证明的方法很多，但都是十分繁杂，我们这里介绍一种最简便的方法。由于梅森公式是求输入信号和输出信号之间的传输值，故首先令：

$$\frac{x_n}{x_0}=K$$

方程式（9—7）改写为

$$\begin{aligned}
&(m_{11}-1)x_1+m_{12}x_2+\cdots+m_{1n}x_n+0x_0=0\\
&m_{21}x_1+(m_{22}-1)x_2+\cdots+m_{2n}x_n+0x_0=0\\
&\cdots\cdots\\
&m_{n-1\,1}x_1+\cdots\cdots\cdots\cdots+m_{n-1\,n}x_n+0x_0=0\\
&\qquad\qquad\qquad\qquad -x_n+Kx_0=0\\
&m_{01}x_1+m_{02}x_2+\cdots\cdots\cdots+m_{0n}x_n-x_0=0
\end{aligned}$$

即 $$MX=0 \tag{9—13}$$

式中M是一个$n+1$阶方阵，它对应一个流图，如图9—15所示。

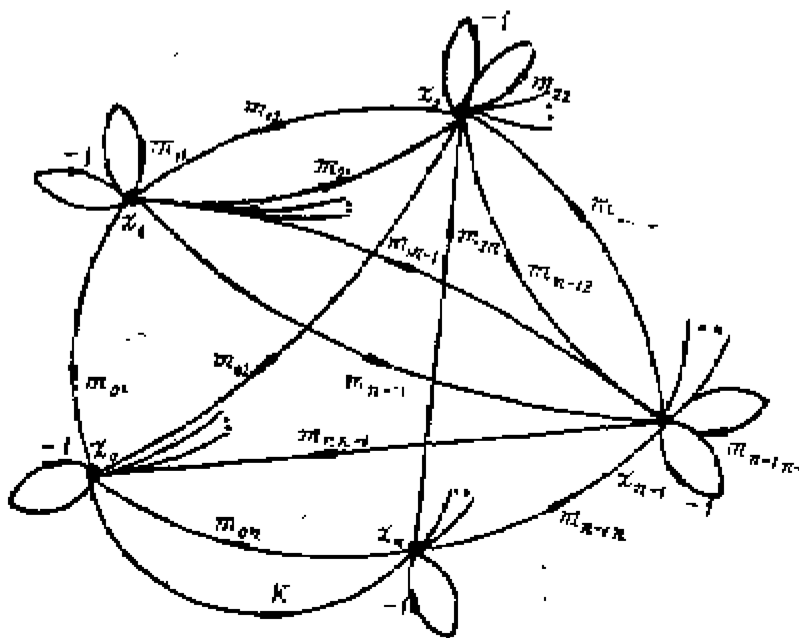


图 9—15

比较图9—15和图9—3发现，同一组方程的信号流图是它的流图的一个子图。换句话说，在信号流图的每个点上，加一个支路传输等于-1的自环，再在输入与输出间加一支路传输等于边k，就得到对应的流图。

为了使方程式（9—12）有非零解，

$$\det M=0$$

即 $$\det M=(-1)^{n+1}\sum(-1)^{k_f}P\text{-组循环乘积}=0。$$

根据P-组循环定义，它是G_f的一个子图。它包括G_f的全部节点，是一个有向回路或若干有向回路的点不共用并集，所以，$\sum(-1)^{k_f}P$-组循环乘积$=\sum(-1)^{k_f}D_f$。对某一边K而言，它等于所有含边K的P-组循环积与所有不含边的P-组循环乘积之和。

令 含有边K的P-组循环积为$D_a(K)$；

不含边K的P-组循环积为$D_i(\overline{K})$。

而　所有含边K的P-组循环积$=\sum(-1)^{k_q}D_q(K)=KP$

所有不含边K的P-组循环积$=\sum(-1)^{k_j}D_j(\overline{K})=C$

则 $$\det M=(-1)^{n+1}[C+KP]=0$$

即 $$C+KP=0$$

$$\frac{1}{K}=\frac{x_0}{x_n}=-\frac{P}{C} \qquad (9-14)$$

式（9—14）清楚地告诉我们

$$\frac{1}{K}=\frac{输出信号}{输入信号}=-\frac{\sum(-1)^{k_q}D_q(K)/K}{\sum(-1)^{k_j}D_j(\overline{K})}$$

$$=-\frac{流图中所有含边k的D_q(K)/K}{流图中所有不含边k的D_j(\overline{K})}$$

由于分子和分母都是不计边K的，当然，利用不含边K的信号流图更为方便了。式（9—14）的分母C是流图中所有不含边K的P-组循环乘积的和，它等于信号流图中的若干有向回路传输$D_j(0)$和若干个(-1)自环$D_j(-1)$的点不共用并集，写成数学式子为

$$C=\sum(-1)^{k_j}D_j(\overline{K})=\sum(-1)^{k_j}[D_j(0)][D_j(-1)]。$$

设第j个P-组循环中有u个自环，则

$$(-1)^{k_j-u}[D_j(0)](-1)^u[\underbrace{(-1)(-1)\cdots(-1)}_{u个}]=(-1)^{k_j-u}D_j(0)$$

令：$k_j'=k_j-u=D_j(0)$中有向回路的个数。

于是　　第j个P-组循环$=\sum(-1)^{k_j'}D_j(0)$

$D_j(0)$是信号流图中一个有向回路传输的点不共用并集。它不是通过信号流图的所有点的。

现假设C_r是信号流图中，那些含有r个有向回路和k_j-r个(-1)自环的P-组传输积之和。

$$C_r=(-1)^r D_j(0)(-1)^{k_j-r}D_j(-1)$$
$$=\sum(-1)^r D_j(0)(-1)^{k_j-r}[\underbrace{(-1)(-1)\cdots(-1)}_{k_j-r个}]=\sum(-1)^r D_j(0)$$

当　$r=0$时，$C_r=1$；

$r=1$和奇数时，$C_r=-\sum D_j(0)$；

$r=2$和偶数时，$C_r=\sum D_j(0)$。

因此 $$C=1-C_1+C_2-\cdots$$

C_1=所有有向回路传输之和。

C_2=所有含二个点不共用有向回路传输之和。

……

C_r=所有含r个点不共用有向回路传输之和。

分子P的每一项是流图中含边k的P-组循环除去k以外的支路传输乘积，它就是信号流图中从输入点到所求输出点的一条有向路径和回路的点不共用并集的传输。

设P中某一项p是流图中一个含边K的P-组循环，不计K的传输。但是这个P-组循环是含有边K和源点到收点的一条路径P_i以及k个点不共用有向回路。因此

$$p=\frac{1}{K}(-1)^{k+1}KP_i\ (k\text{个点不共用回路}) \tag{9—15}$$

式中（k个点不共用回路）不含边K，与点p_i不共用。所以它就是信号流图中删去路径P_i上各点和连接在这些点上的支路后留下的子图$g(P_i)$中的点不共用回路。将（k个点不共用回路）乘以$(-1)^k$就得$C(\overline{P}_i)$，而后者是$g(P_i)$的C，也就是$g(P_i)$的P-组循环积。它包含有$n+1-n_{p_i}$个点，其中n_{p_i}是路径P_i上的点数，$C(\overline{P}_i)$和P_i是点不共用的。因此

$$p=(-1)P_i\cdot C(\overline{P}_i)$$

因为
$$P=\sum p$$

所以
$$-P=\sum P_i\cdot C(\overline{P}_i) \tag{9—16}$$

关于梅森公式，可以总结为下一定理：

定理9—1：设P_i是源点到收点的第i条有向路径的增益，则流图的增益$\frac{x_s}{x_e}$是

$$\frac{1}{K}=\frac{\sum P_i\cdot C(\overline{P}_i)}{C}。 \tag{9—17}$$

例9—3　求图9—16中收点x_3与源点x_0之间传输值。

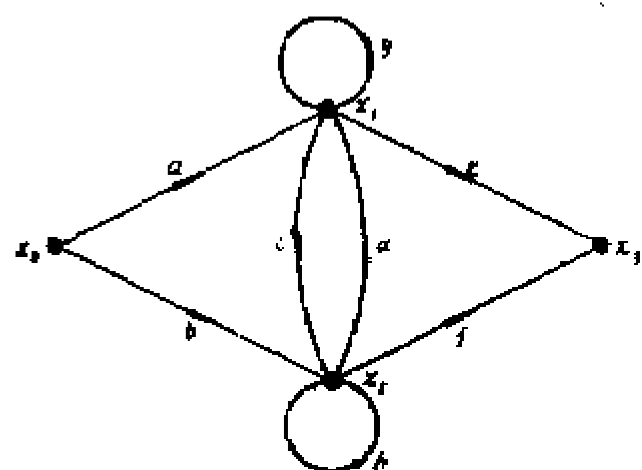


图　9—16

有向回路的传输　表9—1

有向回路	回路传输
x_1自环	g
x_2自环	h
x_1x_2回路	cd

（一）利用梅森公式（9—17）解：

$$\frac{x_0}{x_3}=\frac{1}{K}=\frac{\sum P_i\cdot C(\overline{P}_i)}{C}$$

根据信号流图，共得3个有向回路，它的对应的传输列表如表9—1；

所以
$$e_1=g+h+cd$$
$$e_2=gh$$
$$e_3=C\cdots=0$$
$$e=1-(g\mp h+cd)+gh$$

信号流图中由x_3到x_0的有向路径和对应的$g(P_i)$和$C(\overline{P}_i)$如表9—2所示

因此
$$\frac{x_0}{x_3}=\frac{1}{K}=\frac{ae(1-h)+bf(1-g)+acf+bde}{1-(g+h+cd)+gh}$$

（二）利用梅森公式（9—14），这个公式是对应流图而言的，所以首先要作出流图如图9—17。

$$C=\sum(-1)^{k_i}D_i(\overline{K})$$

$$=(-1)^4(-1)(-1)(-1)(-1)+(-1)^4(-1)(-1)gh$$
$$+(-1)^4(-1)(-1)(-1)g+(-1)^4(-1)(-1)(-1)h$$
$$+(-1)^3(-1)(-1)cd$$

有向路径的g(P_i)和C($\overline{P}_i$) 表9—2

有向路径	路径传输	$g(P_i)$	$C(\overline{P}_i)$
P_1	ae	x_2 h	$1-h$
P_2	bf	x_1 g	$1-g$
P_3	acf	/	1
P_4	bde	/	1

$$=1-g-h-cd+gh$$
$$P=\sum(-1)^{k_i}D_q(K)/K$$
$$=(-1)^1acf+(-1)^1bde+(-1)^2ach$$
$$+(-1)^2bfg+(-1)^2ae(-1)+(-1)^2bf(-1)$$
$$=-acf-bde+ach+bfg-ae-bf$$
$$\frac{x_0}{x_3}=\frac{1}{K}=-\frac{P}{C}=\frac{acf+bde+ae+bf-ach-bfg}{1-(g+h+cd)+gh}$$
$$=\frac{ae(1-h)+bf(1-g)+ach+bde}{1-(g+h+cd)+gh}$$

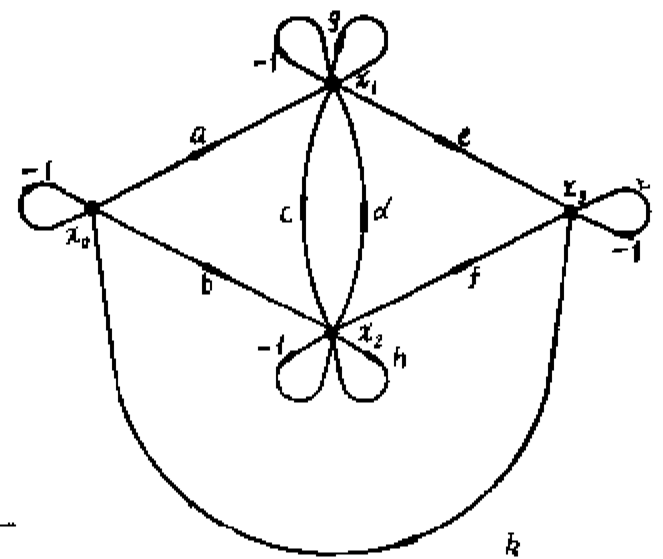


图 9—17

习 题

9—1 用信号流图法解下列线性方程组：

$$\begin{cases}2x+4y=100\\3x-y=1\end{cases}$$

9—2 运用简化规则，求图9—18中点x_0到点x_4的支传输。

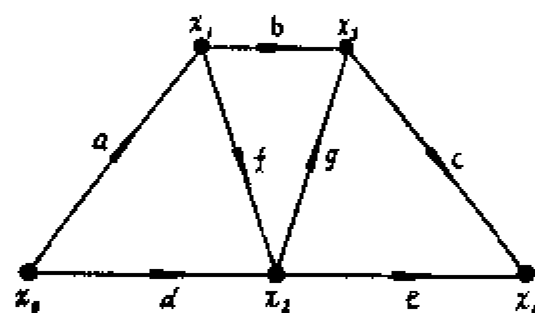


图 9—18

9—3　作图 9—19中网络的信号流图。

9—4　用梅森公式求图 9—20所示信号流图中的x_4/x_0。

9—5　画出图 9—21的信号流图并求$V_2(s)/V_s(s)$和$I_1(s)/V_s(s)$。

9—6　分别用梅森和考推斯公式求图 9—22桥T型网络中输出电压V_o。

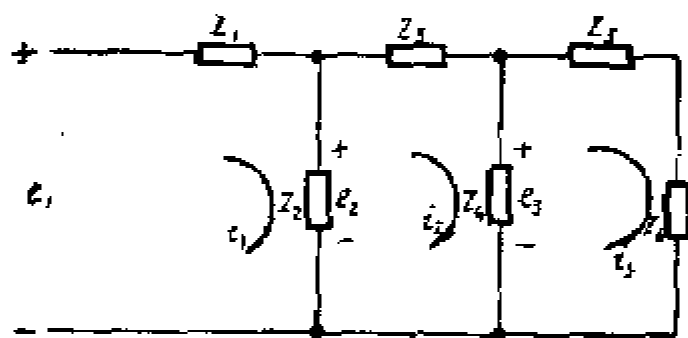


图　9—19

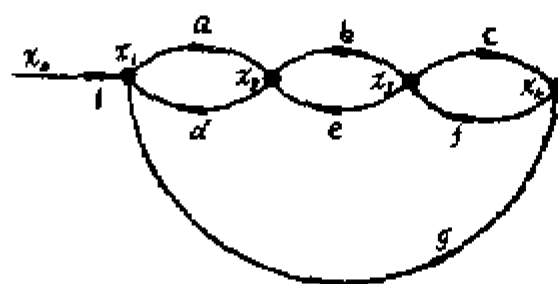


图　9—20

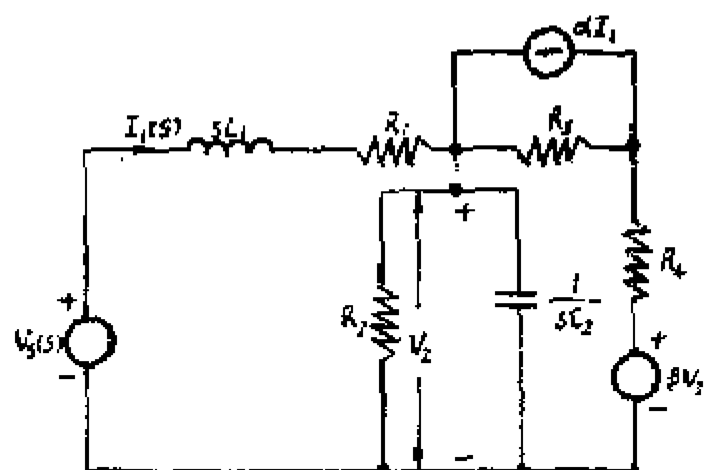


图　9—21

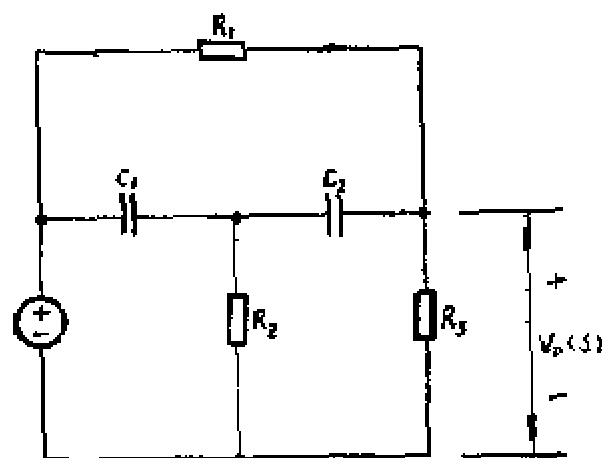


图　9—22

第十章 开关网络

第一节 基本概念

开关网络不仅用在通信系统，如电话交换机，也用在铁路行车自动控制系统，如联锁电路，尤其是近二三十年，还用在数字计算机的设计中。

开关网络中所用的信号是与其它电网络中所用的信号不同的。开关网络中的信号一般只有两个值，即"有"和"无"，或"通"和"断"，常用1和0来表示。这就是二进制信号。开关网络是用来处理和存储二进制信号的。

开关网络可分为两类：即组合网络和时序网络。一个组合网络在某一个给定时间的输出只根据在那个时候的输入；可是，在一个时序开关网络中，在某一给定时间的输出不仅根据那个时候的输入，还要根据整个过去的历史。换句话说，一个时序网络是有记忆的，而组合网络是没有的。在铁路行车自动控制系统中，这两类网络都有应用之处。

开关网络可以看作是一个线性图来处理。在线性图中，我们可以得到开关网络的若干重要拓扑特性，按照这些特性，我们可以进行开关网络的分析和合成。

在铁路行车自动控制和通信系统的开关网络中，一般都是用继电器接点来组成的，所以也可以叫做接点网络。继电器接点是一种双通元件，就是说电流可以沿着两种方向流通。接点组合网络可以用无向图来表示，图中的边是接点，图中的点是接点的引出端子。所以在图论中，一个接点两络是一个没有自环的无向连通图G，每边中有一个二进制变数x_i，它只有两个值：1或0。当接点闭合时是1，而接点断开时是0。

一个接点网络的输入输出关系常用一个二进制变数的函数来表示，如

$$F_i(x_1,\ x_2,\ \cdots,\ x_k)$$

这个函数F_i叫作开关函数或布尔函数，它的值只有1和0。处理开关函数的数学叫作布尔代数或开关代数。

布尔代数对一个有限组x_1，x_2，…，x_k有两种二进制运算，即布尔加法，用符号+和布尔乘法，用符号·。它们要满足下列条件：

（一）x_i或为1，或为0。

（二）每一个变数x_i另有一个变数$\overline{x_i}$，叫作x_i的补数，如果$x_i=0$，则$\overline{x_i}=1$，如果$x_i=1$，则$\overline{x_i}=0$。

（三）1. $x_i+x_j=0$，如果$x_i=x_j=0$；

$x_i+x_j=1$，不然的话。

2. $x_ix_j=1$，如果$x_i=x_j=1$；

$x_ix_j=0$，不然的话。

有了这些简单的公理，我们就可以推导出若干有趣的结果，它们对于开关表达式的简化是很有用的。例如$x_i+x_ix_j=x_i$就是其中之一。

为了研究铁路行车自动控制系统，可以采用系统分析方法。对于一个系统，由于研究的

目的不同，其模型和分析方法也会跟着发生变化。在深入研究系统时，对于同一个系统，会产生不同的模型及分析方法。我们从图10—1可看出系统分析的过程。

铁路行车自动控制的开关网络属结构模型，经过抽象，它可以成为图论中的拓扑数学模型。

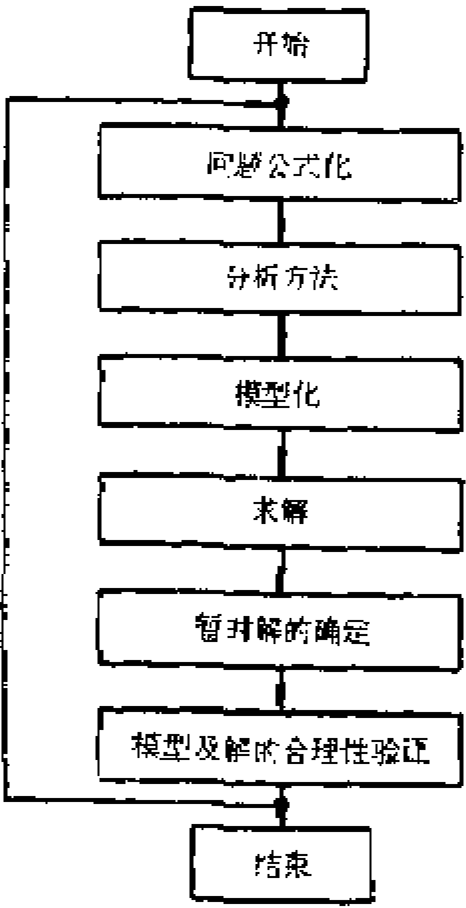


图 10—1

例如64-F型继电半自动闭塞中的发车接收器电路如图10—2所示：

图中J_1、J_2、J_3为继电器，字母$A \sim L$为继电器接点。

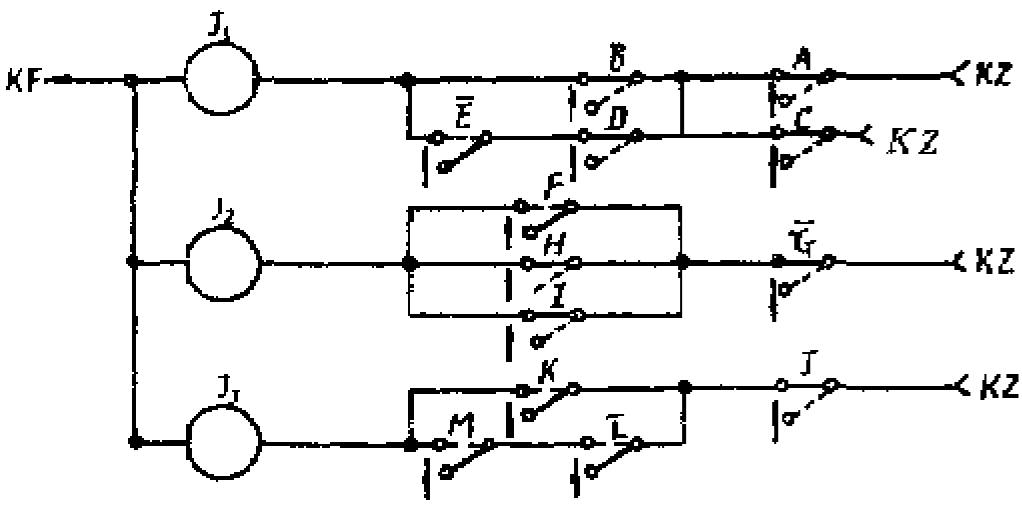


图 10—2

这个图可以抽象为一个网络图，如图10—3所示。图中虚线框内为图论中的开关网络。

铁路电气集中联锁电路是开关网络的一种形式，它的结构也能由图论中的一个线性图来描述。这样的图通常含有较多的点，而且电路图的组成元素也多种多样，因而比较复杂，但它是一个连通图。因此能够用图论的方法来研究网络结构的性质。例如图10—4，这是一个双按钮选岔电路图。在这种电路中，有时会出现迂回电路。为了研究它的结构，解决寻找迂回电路的问题，我们可以把该电路图抽象为一个对应的拓扑几何图，如图10—5所示，这就形成了图论中的图。图10—5中的点号对应着图

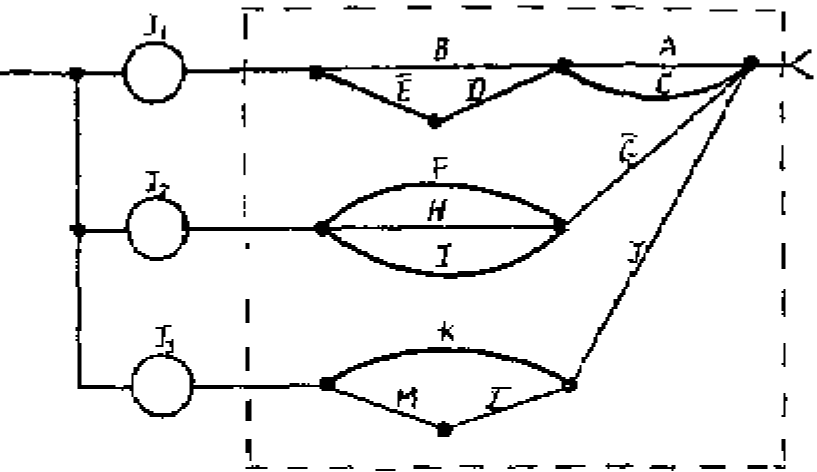


图 10—3

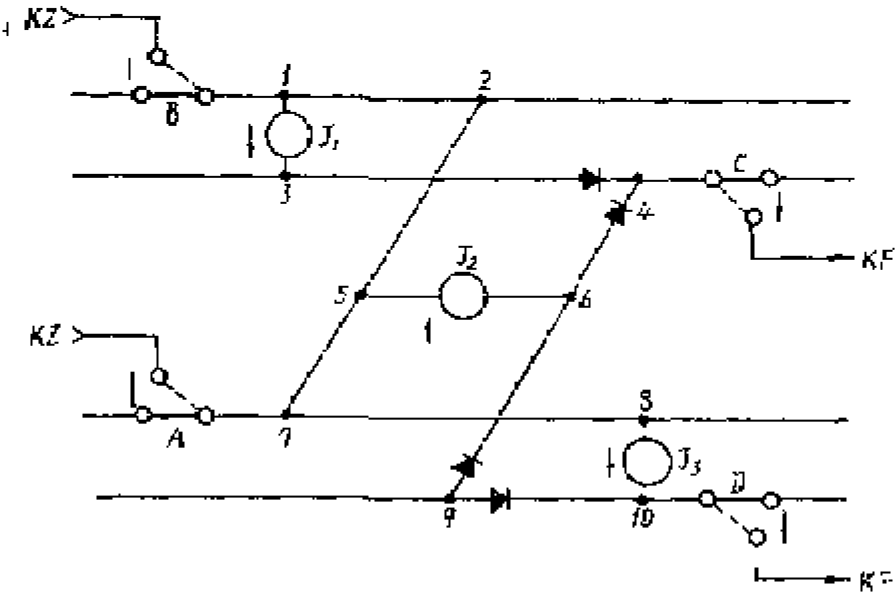


图 10—4

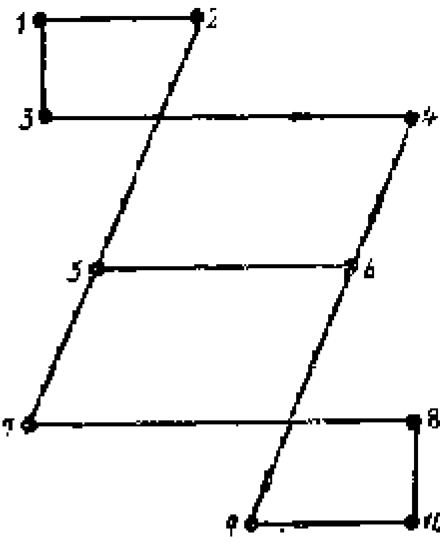


图 10—5

10—4 中的号码。

在原电路中含有二极管，这是一种单向元件。因而在图10—5 中用相应的有向边来表示。在这个图中，既有无向边，又有有向边，因而也称为混合图。

第二节 接通矩阵

人们对于接点网络所关心的是线性图G中某一对点之间的传输情况，就是说，某一对点间是导通还是断开。这最好是用G的一个$n_v \times n_v$矩阵，叫作全接通矩阵或传输矩阵T来表示，即$T=[t_{ij}]$，其中n_v是G的点数，t_{ij}是G中点 i 和 j 之间的传输情况，即两点之间的所有路径。矩阵T是一对称矩阵，而每个对角元素值$t_{ii}=1$。

图10—6 所示的接点网络的传输矩阵T为

$$
\begin{array}{c|cccc|}
 & 1 & 2 & 3 & 4 \\
1 & 1 & x_1(x_2+x_3\overline{x_4}) & x_1(x_3+x_2\overline{x_4}) & x_1 \\
2 & x_1(x_2+x_3\overline{x_4}) & 1 & \overline{x_4}+x_2x_3 & x_2+x_3\overline{x_4} \\
3 & x_1(x_3+x_2\overline{x_4}) & \overline{x_4}+x_2x_3 & 1 & x_3+x_2\overline{x_4} \\
4 & x_1 & x_2+x_3\overline{x_4} & x_3+x_2\overline{x_4} & 1
\end{array}
$$

对于如图10—6 所示那样简单的图来说，列出一个传输矩阵是很容易的。但是当图G比较复杂时，计算所有路径的工作就不是一件轻而易举的事了。所以决定一个图的接通情况的一个比较好的方法是找出一个图的原始接通矩阵Q。

一个有n_v个点的接点网络的原始接通矩阵Q是一个$n_v \times n_v$方矩阵，它的定义为

$$Q=[q_{ij}] \tag{10—1}$$

式中 $q_{ii}=1$；

$q_{ij}=0$，如果点 i 和 j 不是直接由一个接点（即加权边）所连接的；

$q_{ij}=$所有直接连接 i 和 j 之间各边的变数的布尔和。

例如，图10—6 的原始接通矩阵为

$$
Q=\begin{array}{c|cccc|}
 & 1 & 2 & 3 & 4 \\
1 & 1 & 0 & 0 & x_1 \\
2 & 0 & 1 & \overline{x_4} & x_2 \\
3 & 0 & \overline{x_4} & 1 & x_3 \\
4 & x_1 & x_2 & x_3 & 1
\end{array}
$$

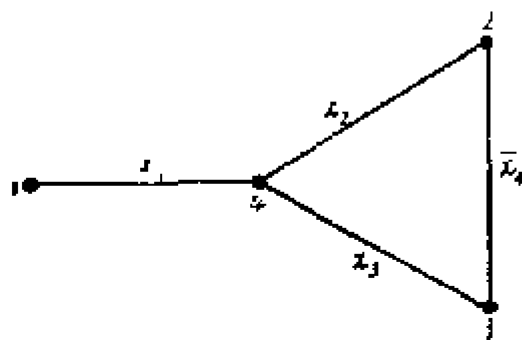


图 10—6

原始接通矩阵是对称的，它包含着一个接点网络的全部信息。

设Q^k是Q的 k 次布尔幂(就是应用布尔代数的规则，Q自乘 k 次)，其中 k 是一个正整数。将Q^k中各元素用下列布尔恒等式简化

$$x+1=x+\overline{x}=1$$

$$x\overline{x}=0$$

$$x+x=xx=x+xy=x$$

然后在简化了的Q^k中检查 i，j 处的元素值，这就是在点 i 和 j 之间总计所有段数为1，2，…，k 的边序列数。我们用这种简化方法消除掉所有冗余项，包括通过同一条边不止一次的（$xx=x$），以及通过同一个点不止一次的（$x+xy=x$）。因此在矩阵Q^k的 i，j 处的元素值代表点 i 和 j 之间有 k 段或少于 k 段的所有路径。因为在一个n_v个点的图中，最长路径的段数为n_v-1，我们得到下一定理。

定理10—1：当一个有n_v个点的接点网络的原始接通矩阵为Q时，它的全接通矩阵T为

$$T=Q_v^{n-1} \qquad (10—2)$$

要是我们只想求取某一对指定点之间的开关函数时，因为定理10—1是求取所有$n_v(n_v-1)/2$对点之间的开关函数的，用了这条定理就有些小题大做了。下一条定理更为有效些。

定理10—2：设Δ_{ij}是原始接通矩阵（用布尔代数运算的，并用布尔恒等式简化的）ij处的余子式（即将行列式删除行 i 和列 j），那么开关函数F_{ij}等于Δ_{ij}。

例10—1　图10—7中开关网络的原始接通矩阵Q为

$$Q=\begin{array}{c|cccc|}
 & 1 & 2 & 3 & 4 \\
1 & 1 & 0 & a & b \\
2 & 0 & 1 & c & d \\
3 & a & c & 1 & e \\
4 & b & d & e & 1
\end{array}$$

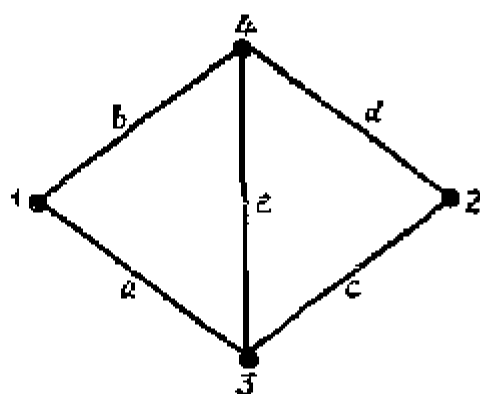


图 10—7

其中a，b，c，d和e代表一些开关函数，也就是各边的权

$$\Delta_{12}=\begin{vmatrix} 0 & c & d \\ a & 1 & e \\ b & e & 1 \end{vmatrix}=ac+ade+bce+bd=F_{12}$$

这就是点1和2之间的开关函数F_{12}。这个开关函数表达成若干个乘积项的和的方式，其中每一个乘积项是几个变数或它们的补数的相乘项，而且每一乘积项中每一变数只出现一次。这种表达方式叫作标准和式。

如果我们取图10—7中能分离点1和2的所有截割组，那么我们将得到另一种表达方式的开关函数F_{12}：

$$F_{12}=(a+b)(c+d)(a+d+e)(b+c+e)$$

这个开关函数表达成若干个和项的乘积，其中每一个和项是几个变数或它们的补数的和，而且每一和项中每一变数只出现一次，这种表达方式叫作标准乘积式。如果将这标准乘积式展开简化，我们就得到上面的标准和式开关函数。

对于复杂的图计算余子式Δ_{ij}是十分麻烦的工作。一个求取F_{ij}的更为简单的方法叫作节点消除法。

为了讲解节点消除法，我们先要介绍几个术语。

在图10—6中，我们可以把点1看作是输入端，点2和3看作是输出端。那么T的1-2元素值说明输入端1与输出端2之间的传输情况。同样，两个输出端之间的传输情况可由T的2-3元素值说明。

如果在矩阵表示法中某些点的情况不是研究对象，那么某些行与列可以不必写入矩阵

中，这种比全接通矩阵的行数和列数较少的矩阵就叫作接通矩阵N。例如，在图10－6所示的图G中，如果点4在矩阵中是不需要的，那么下列的矩阵就是图G的接通矩阵N：

$$N=\begin{matrix} & 1 & 2 & 3 \\ 1 & 1 & x_1(x_2+x_3\overline{x_4}) & x_1(x_3+x_2\overline{x_4}) \\ 2 & x_1(x_2+x_3\overline{x_4}) & 1 & \overline{x_4}+x_2x_3 \\ 3 & x_1(x_3+x_2\overline{x_4}) & \overline{x_4}+x_2x_3 & 1 \end{matrix}$$

对于体现一个函数f的二端网络，如只需要表示输入端1与输出端2的关系，那么有关这个网络的矩阵只是

$$\begin{matrix} & 1 & 2 \\ 1 & 1 & f \\ 2 & f & 1 \end{matrix}$$

这个矩阵叫作这个二端网络的输出矩阵。一般的多端网络，如果在矩阵中只用网络的输入端和输出端，这个矩阵就叫作这个网络的输出矩阵。

凡是一个矩阵中的元素是布尔代数中的一个布尔函数时，这种矩阵叫作布尔矩阵。上述的全接通矩阵，原始接通矩阵，接通矩阵和输出矩阵就是布尔矩阵。另外，这些矩阵都是对称矩阵，它们的对角元素都是1，所以叫作开关矩阵。

节点消除法可以用于一个接通矩阵N（这个矩阵可以是网络的原始接通矩阵或是别的接通矩阵）。假设我们希望消除接通矩阵N的点k，消除的步骤是这样的：我们对N的每一个元素值n_{ij}，其中i，$j\neq k$，和$i\neq j$，加一项n_{ik}、n_{kj}成为n'_{ij}，即$n'_{ij}=n_{ij}+n_{ik}\cdot n_{kj}$。对于每一个$n_{ij}$完成这一步后，我们删除$N$中的行$k$和列$k$，得到一个新的接通矩阵$N_k$。$N_k$是一个接通矩阵，这是因为事实上项$n_{ik}\cdot n_{kj}$代表着在点$i$和$j$之间且其中又经过点$k$的任何可能的联系。照上法继续消除所不需要的点，一直到最后只留下点i和j，我们就得到了点i和j之间的开关函数F_{ij}。

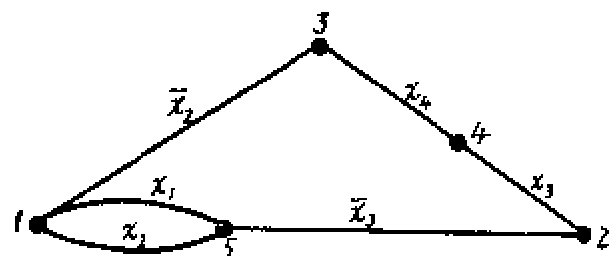


图 10—8

例10－2　在图10－8所示的图中，如果我们要求取点1和点2之间的开关函数F_{12}，那么我们只要写出只包括点1、2和3的接通矩阵如下：

$$N=\begin{matrix} & 1 & 2 & 3 \\ 1 & 1 & \overline{x_3}(x_1+x_2) & \overline{x_2} \\ 2 & \overline{x_3}(x_1+x_2) & 1 & x_3x_4 \\ 3 & \overline{x_2} & x_3x_4 & 1 \end{matrix}$$

为了消除点3，对n_{12}加一项$n_{13}\cdot n_{32}$，得

$$N_3=\begin{matrix} & 1 & 2 \\ 1 & 1 & \overline{x_3}(x_1+x_2)+\overline{x_2}x_3x_4 \\ 2 & \overline{x_3}(x_1+x_2)+\overline{x_2}x_3x_4 & 1 \end{matrix}$$

所以$F_{12}=F_{21}=\overline{x_3}(x_1+x_2)+\overline{x_2}x_3x_4$

例10－3　求图10－9中点1，2，3，4中每两点之间的开关函数，即4×4输出矩

阵。先写出图10—9的原始接通矩阵Q，再以Q_7代表消除点7后所得的矩阵。Q_{76}为从Q_7中消除点6后所得的矩阵，余类推。这些矩阵中所有的元素值已于进行节点消除中加以简化。

$$Q=\begin{array}{c}\\1\\2\\3\\4\\5\\6\\7\end{array}\begin{array}{c}\begin{array}{ccccccc}1&2&3&4&5&6&7\end{array}\\\left[\begin{array}{ccccccc}1&0&x_1&0&\overline{x}_1&x_2&0\\0&1&0&0&\overline{x}_3&x_4&0\\x_1&0&1&x_2&0&0&x_3+x_4\\0&0&x_2&1&0&x_4&\overline{x}_4\\\overline{x}_1&\overline{x}_3&0&0&1&x_3&0\\x_2&x_4&0&x_4&x_3&1&0\\0&0&x_3+x_4&\overline{x}_4&0&0&1\end{array}\right]\end{array}$$

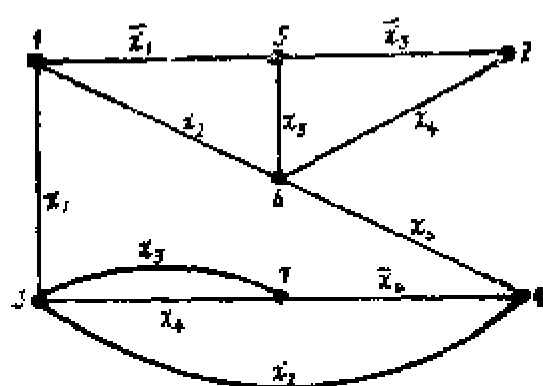


图 10—9

$$Q_7=\begin{array}{c}\\1\\2\\3\\4\\5\\6\end{array}\begin{array}{c}\begin{array}{cccccc}1&2&3&4&5&6\end{array}\\\left[\begin{array}{cccccc}1&0&x_1&0&\overline{x}_1&x_2\\0&1&0&0&\overline{x}_3&x_4\\x_1&0&1&x_2+x_3\overline{x}_4&0&0\\0&0&x_2+x_3\overline{x}_4&1&0&x_4\\\overline{x}_1&\overline{x}_3&0&0&1&x_3\\x_2&x_4&0&x_4&x_3&1\end{array}\right]\end{array}$$

$$Q_{76}=\begin{array}{c}\\1\\2\\3\\4\\5\end{array}\begin{array}{c}\begin{array}{ccccc}1&2&3&4&5\end{array}\\\left[\begin{array}{ccccc}1&x_2x_4&x_1&x_2x_4&\overline{x}_1+x_2x_3\\x_2x_4&1&0&x_4&\overline{x}_3+x_4\\x_1&0&1&x_2+x_3\overline{x}_4&0\\x_2x_4&x_4&x_2+x_3\overline{x}_4&1&x_3x_4\\\overline{x}_1+x_2x_3&\overline{x}_3+x_4&0&x_3x_4&1\end{array}\right]\end{array}$$

$$Q_{765}=\begin{array}{c}\\1\\2\\3\\4\end{array}\begin{array}{c}\begin{array}{cccc}1&2&3&4\end{array}\\\left[\begin{array}{cccc}1&x_2x_4+\overline{x}_1(\overline{x}_3+x_4)&x_1&x_2x_4+\overline{x}_1x_3x_4\\x_2x_4+\overline{x}_1(\overline{x}_3+x_4)&1&0&x_4\\x_1&0&1&x_2+x_3\overline{x}_4\\x_2x_4+\overline{x}_1x_3x_4&x_4&x_2+x_3\overline{x}_4&1\end{array}\right]\end{array}$$

Q_{765}是这个矩阵的一个接通矩阵，但是不一定是一个输出矩阵。为了得到1，2，3和4中每两点之间的开关函数，我们必须求取Q_{765}的幂，一直到$Q_{765}^{n}=Q_{765}^{n+1}$为止。先求$Q_{765}^{2}$，得

$$
Q^2_{765}=\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c}
\begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\
\left[\begin{array}{cccc}
1 & x_2x_4+\overline{x}_1(\overline{x}_3+x_4) & x_1+x_2x_4 & x_1(x_2+x_3\overline{x}_4)+x_4(\overline{x}_1+x_2) \\
x_2x_4+\overline{x}_1(\overline{x}_3+x_4) & 1 & x_2x_4 & x_4 \\
x_1+x_2x_4 & x_2x_4 & 1 & x_2+x_3\overline{x}_4 \\
x_1(x_2+x_3\overline{x}_4)+x_4(\overline{x}_1+x_2) & x_4 & x_2+x_3\overline{x}_4 & 1
\end{array}\right]
\end{array}
$$

其次再求Q^3_{765}，得到$Q^3_{765}=Q^2_{765}$，所以Q^2_{765}是这个网络所需要的输出矩阵。

下面这一定理指出一个开关矩阵是一个输出矩阵的条件。

定理10—3：当且仅当一个接通矩阵N的$N^2=N$时，N是一个双向开关网络的输出矩阵。

上面讲的都是关于双向开关网络的接通矩阵。所以图中没有自环，因为自环对于双向开关网络是没有意义的。对于有向图呢，我们必须注意边的定向，所以有向图的接通矩阵N的定义为

$$N=[n_{ij}] \qquad (10—3)$$

式中　$n_{ij}=\sum\limits_{t=1}^{k}e_t$ ——如果边e_1，e_2，…，e_k是点i和j之间的各边，而且定向都是从i到j的；

$n_{ij}=0$ ——如果边与点无关联，或者虽有关联，而定向却是从j到i的。

例如图10—10的接通矩阵N为

$$
N=\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c}
\begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\
\left[\begin{array}{cccc}
0 & a & 0 & b \\
0 & 0 & c & 0 \\
d & e & f & 0 \\
0 & c & g+h & 0
\end{array}\right]
\end{array}
$$

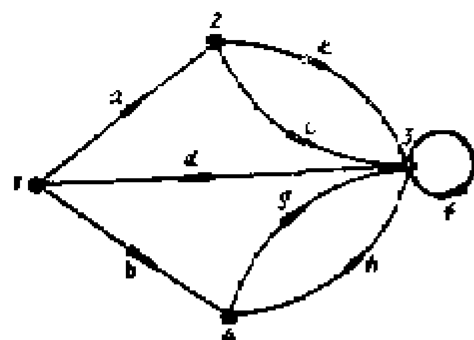


图 10—10

从这里我们可以看出一个有向图的接通矩阵N是与流图的矩阵W很相似的。同时，我们也可以从这个矩阵中看出有向图中各边的位置和定向。为了求取有向图中任何两点之间的开关函数，有下列一条定理。

定理10—4：设一个有向图G的接通矩阵为N，那么从点i到点j的开关函数F_{ij}等于矩阵$[N+U]$ j，i处的余子式Δ_{ji}（用布尔代数运算）。

注意因为在流图中，一条边的权W_{ji}的定向是从点i到j的，所以在有向图G中从点i到点j的开关函数F_{ij}等于矩阵$[N+U]$ j，i处的余子式Δ_{ji}。

例10—4　求图10—10中从点2到点4的开关函数F_{24}。这个网络的接通矩阵N已见上。$[N+U]$为

$$[N+U]=\begin{array}{c} \\ 1\\2\\3\\4\end{array}\begin{array}{c}\begin{array}{cccc}1&2&3&4\end{array}\\ \begin{bmatrix}1&a&0&b\\0&1&c&0\\d&e&1&0\\0&0&g+h&1\end{bmatrix}\end{array}$$

$[N+U]$的余子式Δ_{42}为

$$\Delta_{42}=\begin{vmatrix}1&0&b\\0&c&0\\d&1&0\end{vmatrix}=bcd$$

第三节　开关网络的分析

在上一节中，我们讲到了一个开关网络的各种接通矩阵，并且讲到了利用接通矩阵来求取一个接点网络中任何两点之间的开关函数。在这一节中，我们将用路径集体的性质来分析一个开关网络。

设$\{E\}$是一个线性图G的欧拉图集体，和$\{P_{ij}\}$是G中点i和j之间所有可能路径的集体。由定理2—20知

$$\{P_{ij}\}=\min[P\oplus E \mid E\in\{E\}] \tag{10—4}$$

其中$P\in\{P_{ij}\}$。这就是说，如果我们知道两点之间的一条路径P时，我们就可以利用路径P和G的所有欧拉图的环和来得到所有可能的、在这同样两点之间的所有路径。当G是一个不可分图时，$\{E\}$可以代以$\{C\}$，这个$\{C\}$是G中所有可能回路的集体，即

$$\{P_{ij}\}=\min[P\oplus C\mid C\in\{C\}] \tag{10—5}$$

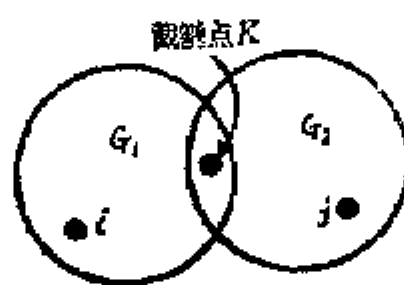


图　10—11

假设G是一个可分图，如图10—11所示，其中G_1和G_2是不可分的，那么很明显，$\{P_{ij}\}$可以表达为

$$\{P_{ij}\}=\{P_{ik}\}\oplus\{P_{kj}\} \tag{10—6}$$

式中$i\in G_1$，$j\in G_2$

$$\{P_{ik}\}=\min\{P_1\ominus C_1\mid C_1\in\{C_1\}\} \tag{10—7}$$

$$\{P_{kj}\}=\min\{P_2\ominus C_2\mid C_2\in\{C_2\}\} \tag{10—8}$$

其中$P_1\in\{P_{ik}\}$，$P_2\in\{P_{kj}\}$和$\{C_1\}$是G_1中所有可能回路的集体，$\{C_2\}$是G_2中所有可能回路的集体。

概括起来，有下面这条定理。

定理10—5：假设一个可分图G可由截割点V_2，V_3，…，V_k分割成k个部分G_1，G_2，…，G_k（$k'\leqslant k$）。再假设在点i和j之间有一条路径P_{ij}通过截割点V_2，V_3，…，V_m（$m\leqslant k'$）。设$i\in G_1$，G_1和G_2共同有一个截割点V_2，G_2和G_3共同有一个截割点V_3，…，G_{m-1}和G_m共同有一个截割点V_m，以及$j\in G_m$，那么

$$\{P_{ij}\}=\{P_{iv_2}\}\otimes\{P_{v_2v_3}\}\otimes\cdots\otimes\{P_{v_mj}\} \tag{10—9}$$

其中$\{P_{iv_2}\}=\min\{P_1\ominus C_1\mid C_1\in\{C_1\}\}$　　(10—10)

$$\{P_{v_r v_{r+1}}\}=\min\{P_r \oplus C_r \mid C_r \in \{C_r\}\} \tag{10—11}$$

$$\{P_{v_m j}\}=\min\{P_m \oplus C_m \mid C_m \in \{C_m\}\} \tag{10—12}$$

$P_1 \in \{P_{iv_2}\}$，$\{P_r\} \in \{P_{v_r v_{r+1}}\}$，$\{P_m\} \in \{P_{v_m j}\}$，而 $r=2,3,\cdots,m-1$ 和 $\{C_u\}$ 是 G 中所有可能回路以及空组的集体，而 $u=1,2,\cdots,m$。另外，开关函数 F_{ij} 可以表达为

$$F_{ij}=F_{iv_1}F_{v_1 v_2}\cdots F_{v_m j} \tag{10—13}$$

式中

$$F_{iv_1}=\sum_{P_1 \in \{P_{iv_1}\}} \text{路径}P_1\text{乘积} \tag{10—14}$$

$$F_{v_r v_{r+1}}=\sum_{P_r \in \{P_{v_r v_{r+1}}\}} \text{路径}P_r\text{乘积，对于}r=2,3,\cdots,m \tag{10—15}$$

$$F_{v_m j}=\sum_{P_m \in \{P_{v_m j}\}} \text{路径}P_m\text{乘积} \tag{10—16}$$

例10—5　在图10—12中，假设我们在点 i 和 j 之间选择一条路径 P_{ij}，如 $P_{ij}=(a_1, b_1, d_1, d_2)$。这条路径通过截割点 V_2 和 V_3。为了求取开关函数 F_{ij}，我们只要考虑 G_1, G_2 和 G_3。从 G_1、G_2 和 G_3 的所有可能回路和空组的集体 $\{C_1\}$、$\{C_2\}$ 和 $\{C_3\}$，其中

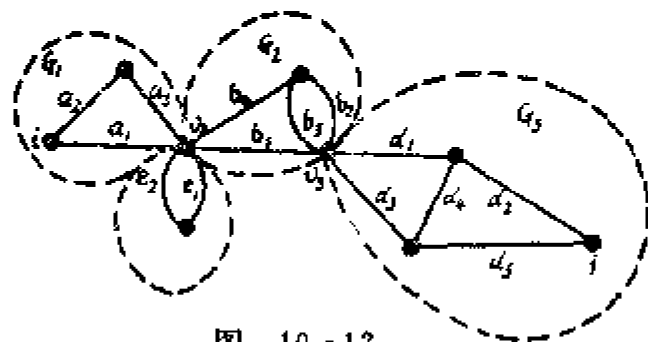


图　10—12

$\{C_1\}=\{\phi,(a_1,a_2,a_3)\}$

$\{C_2\}=\{\phi,(b_1,b_2,b_4),(b_1,b_3,b_4),(b_2,b_3)\}$

$\{C_3\}=\{\phi,(d_1,d_3,d_4),(d_2,d_4,d_5),(d_1,d_2,d_3,d_5)\}$

我们可以得到

$\{P_{iv_2}\}=\min\{(a_1)\oplus C_1 \mid C_1 \in \{C_1\}\}=\{(a_1),(a_2,a_3)\}$

$\{P_{v_2 v_3}\}=\min\{(b_1)\oplus C_2 \mid C_2 \in \{C_2\}\}=\{(b_1),(b_2,b_4),(b_3,b_4)\}$

$\{P_{v_3 j}\}=\min\{(d_1,d_2)\oplus C_3 \mid C_3 \in \{C_3\}\}=\{(d_1,d_2),(d_2,d_3,d_4),(d_1,d_4,d_5),$
$(d_3,d_5)\}$

因此开关函数 F_{ij} 为

$$F_{ij}=F_{iv_2}F_{v_2 v_3}F_{v_3 j}=(a_1+a_2a_3)(b_1+b_2b_4+b_3b_4)(d_1d_2+d_2d_3d_4+d_1d_4d_5+d_3d_5)$$

这个方法适用于计算机分析开关网络。另外，如果需要计算不止一个开关函数时，我们可以应用定理 2 —16中路径集体间的三角关系来简化表达式。

第四节　截割组矩阵的可实现性

关于开关网络的合成，我们在第五章第四节中已经看到一个例子（图 5 — 7）。那是利用回路矩阵的，有时也可以利用截割组矩阵来合成。不过有了回路矩阵或截割组矩阵，并不是一定可以构成线性图的。换句话说，一个二进制的矩阵要成为一个线性图的截割组矩阵或回路矩阵，必须要满足某些条件。

从第五章第八节的式（5—80）或（5—81）中，我们知道从一个截割组矩阵可以得到一个回路矩阵。所以为了研究一个矩阵能否实现为一个图的截割组矩阵或回路矩阵，我们只要研究能否实现为截割组矩阵就够了。另外，一个截割组矩阵总是可以表达为基本截割组矩阵的。所以本节主要研究一个矩阵成为一个图的基本截割组矩阵的可实现性。首先，我们学习一些术语。

如果有一个图它的截割组矩阵为R，那么这个矩阵R就叫作可实现的截割组矩阵。

设矩阵R的形式为$[R_t \vdots U]$，对R的行i而言的一个H-子矩阵，符号为$H(i)$，是R的一个子矩阵，系从G中(1)删除行i中所有非零值的那些列，和(2)删除行i而得到的。例如，假设R为

$$R=\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c}\begin{array}{ccc:cccc} b & d & f & a & c & e & g \end{array} \\ \left(\begin{array}{ccc:cccc} 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right)\end{array}$$

关于行1的H-子矩阵$H_{(1)}$，是把列b和列a连同行1都删除后而得的，即

$$H(1)=\begin{array}{c} \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c}\begin{array}{ccccc} d & f & c & e & g \end{array} \\ \left(\begin{array}{ccccc} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{array}\right)\end{array}$$

为了显示关于行i的一个H-子矩阵$H(i)$是某一个子图的一个截割组矩阵，我们一定先要研究删除一个截割组对一个图的其余截割组的影响。设S_i是一个图G的一个基本截割组矩阵C_f中由行i所代表的一个基本截割组。那么删除行i中所有非零值，即1的各行就等于从G中删除属于截割组S_i的各边。

设V是G中所有点的点组，V_i是属于截割组S_i的有关点的子组，那么$\overline{V}_i=V\ominus V_i$是图$G$中除$V_i$以外的所有其余点的子组。$E(V\times V)$是$G$中所有各边的边组的子图，它就是$G$本身；而$E(V_i\times V_i)$和$E(\overline{V}_i\times\overline{V}_i)$是两个分别与$V_i$和$\overline{V}_i$有关的子图。则

$$E(V\times V)\ominus S_i=E(V_i\times V_i)\cup E(\overline{V}_i\times\overline{V}_i) \qquad (10—17)$$

这表示一个图G删除截割组S_i的各边后，我们得到两个子图$E(V_i\times V_i)$和$E(\overline{V}_i\times\overline{V}_i)$，如图10—13所示。

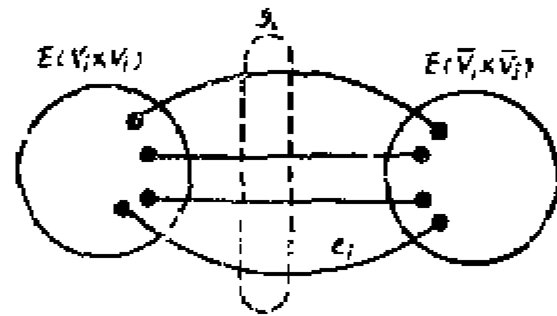


图 10—13

设S_i和S_j是图G的两个基本截割组。当S_i中各边删除后，S_j可能不是G删除S_i后所得到的图G_i的一个基本截割组。

定理10—6′：设$S_i=E(V_i\times\overline{V}_i)$是一个图$G$中关于一棵生成树（或林）$T$的一个基本截割组，并且枝$e_i$是在$S_i$中。设$S_j$是$T$中包括枝$e_j$的另一个基本截割组。假使枝$e_j$是在子图$E(V_i\times V_i)$之中，那么子图$E(V_i\times V_i)$中的一棵生成树$T_1=T\cap E(V_i\times V_i)$中包括枝$e_j$的一个基本截割组$S'_j$将为

$$S'_j=S_j\ominus S_i \qquad (10—18)$$

这里应注意$T\cap E(V_i\times V_i)$是生成树T的一部分，是在子图$E(V_i\times V_i)$之中。

证：很明显，线性图$E(V_i\times V_i)\cup E(\overline{V}_i\times\overline{V}_i)$是分离的，所以在生成树$T$中插入子图$E(\overline{V}_i\times\overline{V}_i)$中的任何一条弦将产生一个回路$C$，它只包括$E(\overline{V}_i\times\overline{V}_i)$中的各边。这就是说枝$e_i$不在回路$C$之中。所以截割组$S_i$不包括$E(\overline{V}_i\times\overline{V}_i)$中任何一条弦。从一个基本截割组的定义可知$S_i$中不包括属于$E(\overline{V}_i\times\overline{V}_i)$的树$T$的任何一枝。所以$S_i$不包括$E(\overline{V}_i\times\overline{V}_i)$中任何一条边。

我们知道当且仅当在$E(V_i\times V_i)$中的一条弦e_p和生成树T_1产生包括枝e_j的一个回路时，弦e_p是在S_j'之中。可是，弦e_p和生成树T将产生同样这个回路。所以，如果弦e_p是在S_j'之中，它也在S_j之中，或者$S_j'\subset S_j$。

既然S_i不包括$E(\overline{V}_i\times\overline{V}_i)$中任何一条边，并且不在$S_j$中的弦一定是一条在$S_i$中的弦，所以$S_j'=S_j\ominus S_i$。这个论证对于$E(\overline{V}_i\times\overline{V}_i)$是空组时也是适用的。▌

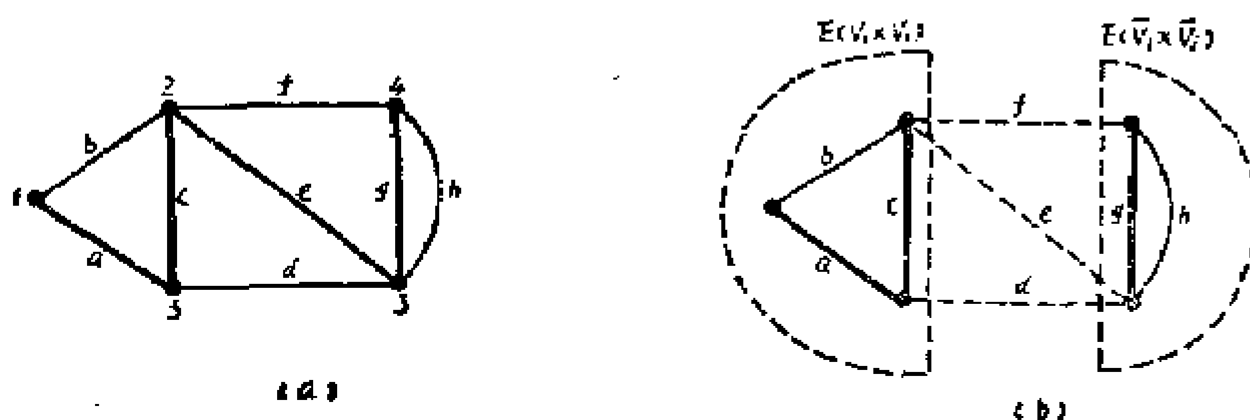


图 10—14

例10—6　在图10—14(a)中的线性图G中，假使我们选择一棵生成树$T=(a,\ c,\ e,\ g)$，一个基本截割组$S_i=E(V_i\times\overline{V}_i)=(d,\ e,\ f)$，其中$V_i=(1,\ 2,\ 5)$，$\overline{V}_i=(3,\ 4)$。从图$G$中删除$S_i$中所有各边后得到的图$E(V_i\times V_i)\cup E(\overline{V}_i\times\overline{V}_i)$如图($b$)所示。我们可以看到$T_1=T\cap E(V_i\times V_i)=(a,\ c)$，是子图$E(V_i\times V_i)$中的一棵生成树。

设另一个基本截割组$S_j=E(V_j\times\overline{V}_j)=(b,\ c,\ d)$，其中$V_j=(1,\ 5)$和$\overline{V}_j=(2,\ 3,\ 4)$。注意$T$中的枝$c$是在$S_j$之中。在$E(V_i\times V_i)$中，对$T_1$而言，只有一条弦$b$，这条弦加到$T_1$将产生一个回路$C=(a,\ b,\ c)$，它包括枝$c$。所以基本截割组$S_j'$对$E(V_i\times V_i)$中的$T_1$而言，是$S_j'=(b,\ c)$，这就是$S_j'=S_j\ominus S_i$。

现在我们可从以下一条定理中看出H—子矩阵$H_{(i)}$是子图$E(V_i\times V_i)\cup E(\overline{V}_i\times\overline{V}_i)$中的一个基本截割组矩阵。

定理10—7：设一个图G中关于一棵生成树T的一个基本截割组矩阵为C_f。设C_f中行i代表基本截割组$S_i=E(V_i\times\overline{V}_i)$。那么关于$C_f$中行$i$的$H$-子矩阵$H_{(i)}$是一个图$E(V_i\times V_i)\cup E(\overline{V}_i\times\overline{V}_i)$中关于生成林$T_1\cup T_2$的一个基本截割组矩阵，其中$T_1=T\cap E(V_i\times V_i)$和$T_2=T\cap E(\overline{V}_i\times\overline{V}_i)$。

证：由定理10—6知道，因为在$E(V_i\times V_i)$中关于生成树T_1的每一个基本截割组S_j'可以表达为$S_j\ominus S_i$。同样，在$E(\overline{V}_i\times\overline{V}_i)$中关于生成树$T_2$的每一个基本截割组$S_k'$可以写作$S_k'=S_k\ominus S_i$。关于$C_f$中行$i$的一个$H$-子矩阵$H_{(i)}$可以由$S_i$中删除对应各弦的各列和删除对应一条枝的行$i$来得到的。所以在$H_{(i)}$中每一行代表$S_p\ominus S_i$，并且在$C_f$中除行$i$外，每行都是在$H_{(i)}$之中。▌

例10—7　在图10—14(a)的图G中，关于生成树$T=(a,\ c,\ e,\ g)$的一个基本截割组矩阵C_f是

$$C_f=\begin{matrix} & \begin{matrix} b & d & f & h & & e & c & a & g \end{matrix} \\ \begin{matrix} i \\ 1 \\ 2 \\ 3 \end{matrix} & \left[\begin{array}{cccc:cccc} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right] \end{matrix}$$

关于行 i 的C_f的H-子矩阵$H(i)$是

$$H_{(i)}=\begin{matrix} & \begin{matrix} b & h & & c & a & g \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \left[\begin{array}{cc:ccc} 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{array}\right] \end{matrix}$$

这是图10—14(b)中图$E(V_i\times V_i)\cup E(\overline{V}_i\times\overline{V}_i)$中关于生成林$T_1\cup T_2=(c,\ a,\ g)$的一个基本截割组矩阵。

假使$E(V_i\times V_i)$和$E(\overline{V}_i\times\overline{V}_i)$都不是空的，那么因为图$E(V_i\times V_i)\cup E(\overline{V}_i\times\overline{V}_i)$的基本截割组可以分成两个集体；一个集体包括关于$T_1$的所有基本截割组，另一个包括关于$T_2$的所有基本截割组。可以看出，两个集体中的基本截割组间是没有共同边的。所以各行代表着这些基本截割组的H-子矩阵$H(i)$可以分块成

$$H_{(i)}=\left[\begin{array}{c:c} H(i)_1 & 0 \\ \hdashline 0 & H(i)_2 \end{array}\right] \tag{10—19}$$

式中$H(i)_1$代表图$E(V_i\times V_i)$中的一个基本截割组矩阵，而$H(i)_2$代表图$E(\overline{V}_i\times\overline{V}_i)$中的一个基本截割组矩阵。当$E(\overline{V}_i\times\overline{V}_i)$是空组时，$H(i)_2$也是空的。

另一方面，当把$H_{(i)}$任意分块成式（10—19）形式时，$H(i)_1$和$H(i)_2$也许不能分别给出子图$E(V_i\times V_i)$和$E(\overline{V}_i\times\overline{V}_i)$的基本截割组矩阵。

例10—8　图10—15(a)的线性图G中关于生成树$T=(e,\ f,\ g,\ h,\ i)$的基本截割组矩阵C_f是

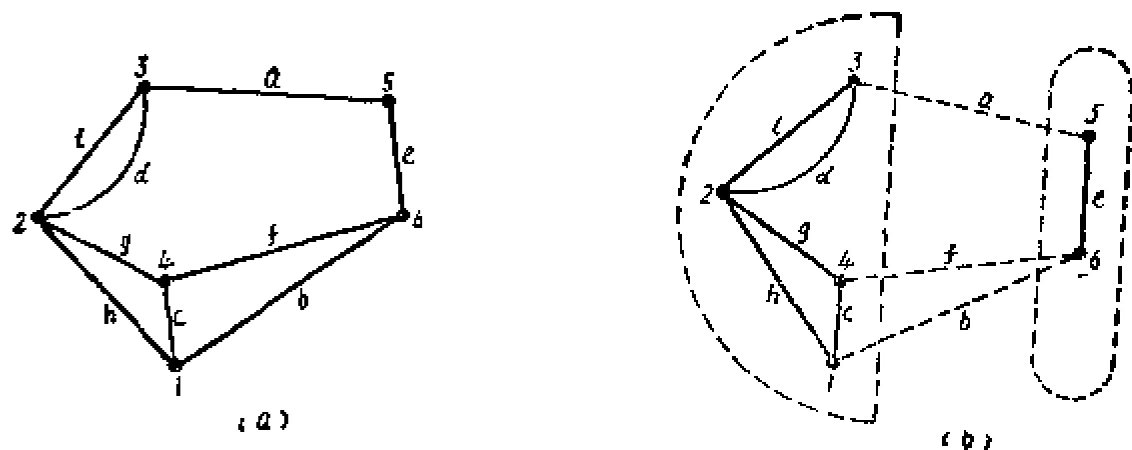


图　10—15

$$
C_f=\begin{array}{c}\\ 1\\2\\3\\4\\5\end{array}\begin{array}{c}\begin{array}{cccc:ccccc} a & b & c & d & e & f & g & h & l\end{array}\\ \left[\begin{array}{cccc:ccccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0\\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0\\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1\end{array}\right]\end{array}
$$

一个H-子矩阵$H(2)$是

$$
H(2)=\begin{array}{c}\\ 1\\3\\4\\5\end{array}\begin{array}{c}\begin{array}{cc:cccc} c & d & e & g & h & l\end{array}\\ \left[\begin{array}{cc:cccc} 0 & 0 & 1 & 0 & 0 & 0\\ 1 & 0 & 0 & 1 & 0 & 0\\ 1 & 0 & 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0 & 0 & 1\end{array}\right]\end{array}
$$

从G中删除基本截割组$S_2=(a, b, f)$中各边后所得的图$E(V_2\times V_2)\cup E(\overline{V}_2\times\overline{V}_2)$，如图10—15($b$)所示。其中$V_2=(1, 2, 3, 4)$和$\overline{V}_2=(5, 6)$。

把$H_{(2)}$分块成：

$$
H(2)=\begin{array}{c}\\ 3\\4\\5\\ \\1\end{array}\begin{array}{c}\begin{array}{ccccc:c} c & d & g & h & l & e\end{array}\\ \left[\begin{array}{ccccc:c} 1 & 0 & 1 & 0 & 0 & \\ 1 & 0 & 0 & 1 & 0 & 0\\ 0 & 1 & 0 & 0 & 1 & \\ \hline & & 0 & & & 1\end{array}\right]\end{array}=\left[\begin{array}{c:c} H(2)_1 & 0\\ \hdashline 0 & H(2)_2\end{array}\right]
$$

这里可以看出$H(2)_1$是$E(V_2\times V_2)$中关于$T_1=(g, h, l)$的基本截割组矩阵，而$H(2)_2$是$E(\overline{V}_2\times\overline{V}_2)$中关于$T_2=(e)$的基本截割组矩阵。另一方面，我们也可以把$H(2)$任意分块为：

$$
H(2)=\begin{array}{c}\\ 5\\ \\1\\3\\4\end{array}\begin{array}{c}\begin{array}{cc:cccc} d & l & c & e & g & h\end{array}\\ \left[\begin{array}{cc:cccc} 1 & 1 & & 0 & & \\ \hdashline & & 0 & 1 & 0 & 0\\ & 0 & 1 & 0 & 1 & 0\\ & & 1 & 0 & 0 & 1\end{array}\right]\end{array}=\left[\begin{array}{c:c} H(2)_1 & 0\\ \hdashline 0 & H(2)_2\end{array}\right]
$$

可是这样分块出来的$H(2)_1$和$H(2)_2$都不是$E(V_2\times V_2)$中关于$T_1=(g, h, l)$的基本截割组矩阵。

下面我们要看看关于生成树T中的一树枝被短捷时对于一个基本截割组矩阵的影响。当我们把这个结果推广，并且连同H-子矩阵的性质，就可以得到矩阵的可实现性测试算法。

定理10—8：设C_f是一个图中关于一棵生成树T的一个基本截割组矩阵。设S_i是C_f中行i所代表的一个基本截割组。设e_i是S_i中的一树枝。再设$C(\overline{i})_f$是C_f的一个子矩阵，得自（1）删除行i，和（2）删除所有零值列。那么$C(\overline{i})_f$是$G(\overline{e_i})$的一个基本截割组矩阵，而$G(\overline{e_i})$是从G中短捷枝e_i后所得到的图。

证：这是因为$T\ominus(e_i)$是$G(\overline{e_i})$的一棵生成树。设e_j是T中不同于e_i的另一枝。假设在树T中插入一条弦产生一个包括e_j的回路C，这个回路当枝e_i短捷后仍是一个包括e_j的回路。所以G中关于T的所有基本截割组，当e_i短捷时，除S_i外，将是关于$T\ominus(e_i)$的所有基本截割组。▮

假设我们要得到一个如图10—16所示线性图G'的一个基本截割组矩阵M，而G'是得自图10—13，不过把子图$E(\overline{V}_i\times\overline{V}_i)$中的所有各点合并在一起。我们知道$E(\overline{V}_i\times\overline{V}_i)$是从图$G$中删除关于生成树$T$的一个截割组$S_i=E(V_i\times\overline{V}_i)$中所有各边后所得到的两个最大连通子图中的一个。我们也知道$T_2=T\cap E(\overline{V}_i\times\overline{V}_i)$是图$E(\overline{V}_i\times\overline{V}_i)$的一棵生成树。因为$T_2$包括$E(\overline{V}_i\times\overline{V}_i)$中所有的点，而$E(\overline{V}_i\times\overline{V}_i)$中的各点可以把$T_2$中所有各枝短捷而使之重合。所以根据定理10—8，一个图G'中关于树$T\ominus T_2$的一个基本截割组矩阵M可以从基本截割组矩阵C_f中删除去代表着包括T_2中各枝的基本截割组的各行，并且删除去所有零值的各列而得到的。

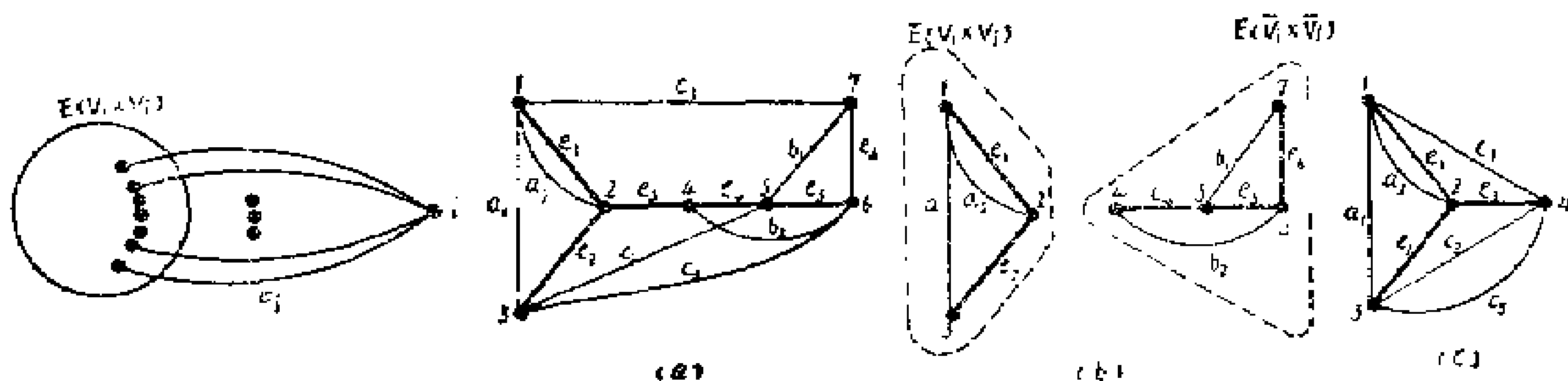


图 10—16　　　　　　图 10—17

例10—9　在图10—17(a)的图G中，关于生成树$T=(e_1, e_2, e_3, e_4, e_5, e_6)$的基本截割组矩阵$C_f$是

$C_f=$

	a	a_2	c_1	c_2	c_3	b_1	b_2	e_1	e_2	e_3	e_4	e_5	e_6
S_{e1}	1	1	1	0	0	0	0	1	0	0	0	0	0
S_{e2}	1	0	0	1	1	0	0	0	1	0	0	0	0
S_{e3}	0	0	1	1	1	0	0	0	0	1	0	0	0
S_{e4}	0	0	1	1	1	0	1	0	0	0	1	0	0
S_{e5}	0	0	1	0	1	1	1	0	0	0	0	1	0
S_{e6}	0	0	1	0	0	1	0	0	0	0	0	0	1

设S_i是$Se_3=E(V_i\times\overline{V}_i)=(C_1, C_2, C_3, e_3)$。所以$V_i=(1, 2, 3)$和$\overline{V}_i=(4, 5, 6, 7)$，$S_{e_3}$的相应子图$E(V_i\times V_i)$和$E(\overline{V}_i\times\overline{V}_i)$如图10—17(b)所示。从这些子图中可以看出$T_1=(e_1, e_2)$是$E(V_i\times V_i)$的一棵生成树，和$T_2=(e_4, e_5, e_6)$是$E(\overline{V}_i\times\overline{V}_i)$的一棵生成树。

在图G中把$E(\overline{V}_i\times\overline{V}_i)$的所有各点都重合在一起而得到的图$G'$，如图10—17(c)所示。我们可以看出$T\ominus T_2=(e_1, e_2, e_3)$是$G'$的一棵生成树，并且关于$T\ominus T_2$的基本截割组是$S_{e_1}$，$S_{e_2}$和$S_{e_3}$。

为了从C_f中得到G'中关于生成树$T\ominus T_2$的一个基本截割组矩阵M，我们删除对应基本截割组S_{e_4}，S_{e_5}和S_{e_6}的各行，结果得

$$
\begin{array}{c|ccccccccccccc|}
 & a_1 & a_2 & c_1 & c_2 & c_3 & b_1 & b_2 & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
2 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
3 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{array}
$$

从这个矩阵，我们删除所有零值各列以得到一个在G'中关于$T\ominus T_2$的基本截割组矩阵M

$$
M=\begin{array}{c|ccccc|ccc|}
 & a_1 & a_2 & c_1 & c_2 & c_3 & e_1 & e_2 & e_3 \\
1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
2 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
3 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1
\end{array}
$$

从例10—9中，我们清楚为了能够得到G'的一个基本截割组矩阵M，必须（1）删除对应于$S_{e'_1}$，$S_{e'_2}$，…，$S_{e'_k}$的各行，其中e'_1，e'_2，…，e'_k系子图$E(\overline{V}_i\times\overline{V}_i)$中生成树$T_2=T\cap E(\overline{V}_i\times\overline{V}_i)$中的各枝，以及（2）删除各零值列。所以我们只需知道C_f中哪些行对应于截割组$S_{e'_1}$，S'_{e_2}，…，$S_{e'_k}$。现在，我们准备应用H-子矩阵的特性。回忆$E(\overline{V}_i\times\overline{V}_i)$的一个基本截割组矩阵是$H(i)_2$分块成式（10—19）的形式。如果$H(i)$可以有不同方法分块成式（10—19）的形式，那么其中有一种分块方法将给出对应于$E(\overline{V}_i\times\overline{V}_i)$的一个基本截割组矩阵。所以为了得到$G'$的一个基本截割组矩阵$M$所需删除的各行就是$H(i)_2$的各行。由于删除这些行后所产生的零值列很明显是对应于$E(\overline{V}_i\times\overline{V}_i)$的各边，所以我们可以说，从$C_f$中删除$H(i)_2$的所有行和列，就可以得到$G'$的$M$。

同样，在图10—13的G中，把$E(V_i\times V_i)$中的各点重合，如图10—18所示。从中得到图G''中关于生成树$T\ominus T_1$的一个基本截割组矩阵，它是可以从C_f中删除$H(i)_1$的所有行和列来得到的。G'和G''的这两个基本截割组矩阵叫作M-子矩阵对。

设R的形式为$[R_1 \mid U]$。对于R中行i的H-子矩阵$H(i)$的固定分块，如式（10—19）所示。R的一个M-子矩阵对是一对子矩阵$M(i)_1$和$M(i)_2$，其中$M(i)_1$系由R中删除属于$H(i)_2$的所有行和列而得到的，$M(i)_2$系由R中删除属于$H(i)_1$的所有行和列而得到的。

注意，当$H(i)_1$是零时，既然从R中没有行和列可以删除，所以，$M(i)_2=R$。同样，如果$H(i)_2$是零时，$M(i)_1=R$。

例10—10　在例10—8中，我们可以得到$H(2)_1$和$H(2)_2$为

$$
H(2)_1=\begin{array}{c|ccccc|}
 & c & d & g & h & l \\
3 & 1 & 0 & 1 & 0 & 0 \\
4 & 1 & 0 & 0 & 1 & 0 \\
5 & 0 & 1 & 0 & 0 & 1
\end{array}
$$

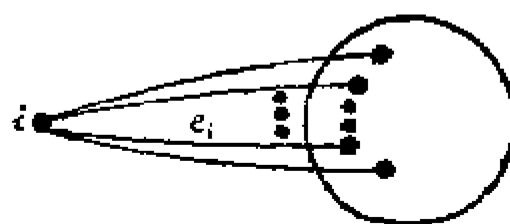


图　10—18

和

$$
H(2)_2=\begin{array}{c|c|}
 & e \\
1 & 1
\end{array}
$$

C_f中关于行2的M-子矩阵对$M(2)_1$和$M(2)_2$可用下列步骤求得。为了求得$M(2)_1$，我们删除行1和列e〔这从C_f中组成$H(2)_2$〕。结果为

$$M(2)_1=\begin{array}{c}\\2\\3\\4\\5\end{array}\begin{array}{c}\begin{array}{cccccccc}a&b&c&d&f&g&h&l\end{array}\\\left[\begin{array}{cccccccc}1&1&0&0&1&0&0&0\\1&1&1&0&0&1&0&0\\0&1&1&0&0&0&1&0\\1&0&0&1&0&0&0&1\end{array}\right]\end{array}$$

因为$H(2)_1$包括行3，4和5，和列c，d，g，h和l，从C_f中删除这些行和列，得$M(2)_2$为

$$M(2)_2=\begin{array}{c}\\1\\2\end{array}\begin{array}{c}\begin{array}{cccc}a&b&e&f\end{array}\\\left[\begin{array}{cccc}1&0&1&0\\1&1&0&1\end{array}\right]\end{array}$$

而

$$C_f=\begin{array}{c}\\1\\2\\3\\4\\5\end{array}\begin{array}{c}\begin{array}{cccc|ccccc}a&b&c&d&e&f&g&h&l\end{array}\\\left[\begin{array}{cccc|ccccc}1&0&0&0&1&0&0&0&0\\1&1&0&0&0&1&0&0&0\\1&1&1&0&0&0&1&0&0\\0&1&1&0&0&0&0&1&0\\1&0&0&1&0&0&0&0&1\end{array}\right]\end{array}$$

子图G'和G''如图10—19所示。

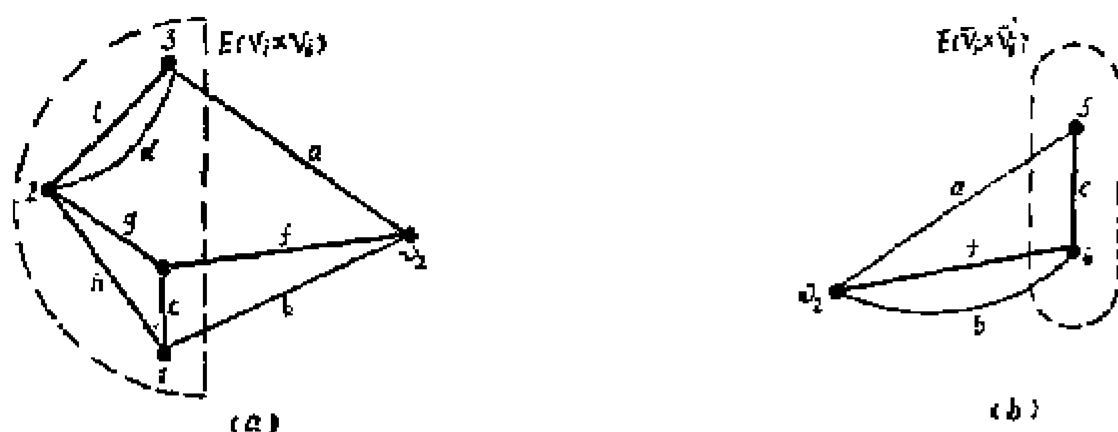


图 10—19

我们可以看出，一个基本截割组矩阵C_f中关于行i的M-子矩阵对的$M(i)_1$和$M(i)_2$有下列特性：

（一）$M(i)_1$和$M(i_2)$是线性图的基本截割组矩阵。

（二）C_f中除行i以外的任何行是在$M(i)_1$之中，或在$M(i)_2$之中，但不是在两者之中。C_f的行i是既在$M(i)_1$之中，又在$M(i)_2$之中。

（三）C_f中除在行i中有1的那些列以外的任何列是在$M(i)_1$之中，或在$M(i)_2$之中，但不在两者之中。在C_f中行i有1的各列是既在$M(i)_1$之中，又在$M(i)_2$之中。

（四）在线性图G'中〔它的基本截割组矩阵是$M(i)_1$，并且它是在$E(\overline{V}_i\times\overline{V}_i)$中重合各点得来的〕，$M(i)_1$中行$i$所代表的基本截割组也是一个关联组。同样，在线性图$G''$中〔它是在$E(V_i\times V_i)$中重合各点得来的〕，$M(i)_2$中行$i$所代表的基本截割组也是一个关联组。

特性（一）、（二）和（三）是从一个基本截割组C_f所组成$M(i)_1$和$M(i)_2$的过程中直接得出的。从特性中我们想起G'是由于$E(\overline{V}_i\times\overline{V}_i)$中各点重合而得来的。所以$C_f$中行$i$所

代表的基本截割组变成G'中的关联组。同样，G''是由$E(\overline{V}_i\times\overline{V}_i)$中各点重合而得来的。所以$C_f$中行$i$所代表的基本截割组变成$G''$中的关联组。

因为我们通过上述步骤总是可以从C_f中得到一对M-子矩阵和相应的图G'和G''，我们有下面的定理。

定理10—9：如果C_f是一个基本截割组矩阵，则对应于C_f中行i有一对M-子矩阵$M(i)_1$和$M(i)_2$，它们有下列特性：

（一）$M(i)_1$是一个线性图的一个基本截割组矩阵，另外，$M(i)_1$中行i所代表的基本截割组是线性图的一个关联组。

（二）$M(i)_2$是一个线性图的一个基本截割组矩阵，其中$M(i)_2$中行i所代表的基本截割组是线性图的一个关联组。

在图10—20(a)中有一个线性图G，其中基本截割组S_i包括边e'_1，e'_2，…，e'_k。假使我们在每条边的中间添加一个点，如图(a)所示。新添这些点后，假使我们把这些点重合成为一个点i，如图(b)所示。现在我们把点i分割为二，结果把图G分离为G'和G''，如图(c)所示。我们看到，图G'和G''是由于把$E(V_i\times V_i)$和$E(\overline{V}_i\times\overline{V}_i)$中的各点分别重合而得到的。

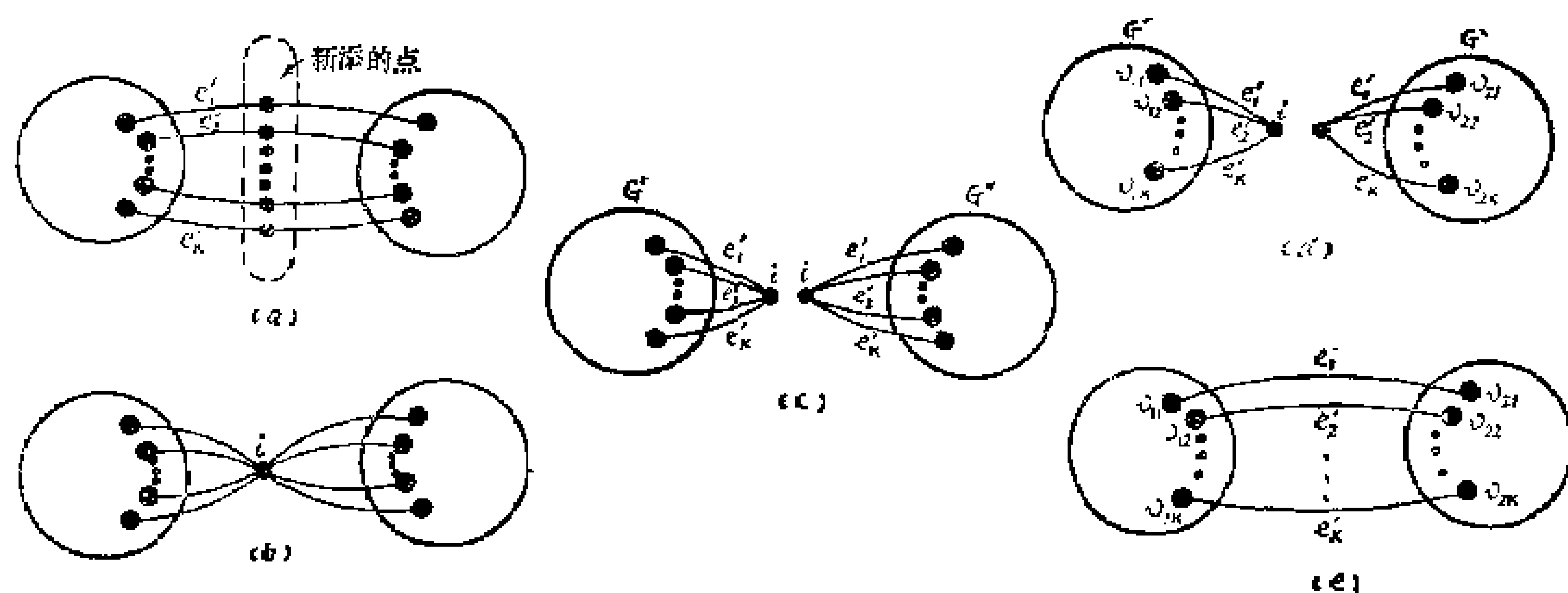


图 10—20

假使图10—20(c)中G'和G''是任何两个线性图，它们的特性有：(1)$M(i)_1$和$M(i)_2$分别为G'和G''的基本截割组矩阵，和(2)G'的$M(i)_1$中行i所代表的基本截割组是一个关联组，和G''的$M(i)_2$中行i所代表的基本截割组是一个关联组。在这些情况下，在G'和G''中的各边是那些由$M(i)_1$和$M(i)_2$的行i所代表的关联组所对应的一个点所连接着的。设这一个点是点i。就是说，G'中点i是$M(i)_1$中行i所代表的关联所对应的点，和G''中点i是$M(i)_2$中行i所代表的关联组所对应的点。设连接到G'和G''中点i的各边为e'_1，e'_2，…，e'_k。又设G'中的点v_{1p}是边e'_p所连接的点，G''中的点v_{2p}是边e'_p所连接的点，其中$p=1$，2，…，k，如图(d)所示。然后将G'和G''中的两条边e'_p用一条边来代替，后一条边连接着v_{1p}和v_{2p}，$p=1$，2，…，k，我们就得到一个线性图G，如图(e)所示。可以看出这个线性图G的基本截割组矩阵是C_f。所以连同定理10—8，我们得到下一定理。

定理10—10：当且仅当一个矩阵C_f中对应于行i的一对M-子矩阵$M(i)_1$和$M(i)_2$可以分别实现为线性图G'和G''的基本截割组矩阵，并且使得$M(i)_1$的行i和$M(i)_2$的行i分别为G'和G''的关联组时，这个矩阵C_f才可以实现为一个线性图G的基本截割组矩阵。

必须十分注意H-子矩阵和M-子矩阵的行名和列名是对应着一个给定矩阵C_f的行和列的。例如，$M(i)_1$的行i对应着C_f的行i的。

正像我们测试一个矩阵C_f的一对M-子矩阵$M(i)_1$和$M(i)_2$来检查C_f是否是一个基本截割组矩阵那样，我们可以测试一个矩阵$M(i)_1$的一对M-子矩阵来检查$M(i)_1$是否是一个基本截割组矩阵。但是有一个约束，就是行i要表示出是一个关联组。此外，我们可以拿这对子矩阵中的一个作为一个给出的矩阵，并且做成另外一对M-子矩阵来检查一个矩阵是否是一个基本截割组矩阵，并且某些特定行表示出是些关联组。这个方法可以继续下去。在进行中，我们应用下述的方法来表示出一些行代表一些关联组。

在一个M-子矩阵的括号内由元素值所表明的行j_1，j_2，…，j_k，如$M(j_1, j_2, \cdots j_k)$一定是代表关联组那些行。上面已经讲过，当两个线性图G'和G''的基本截割组矩阵分别为$M(i)_1$和$M(i)_2$，而要从它们得到一个线性图G时，我们只要把对应着由行i所代表的关联组的点i上所连接着的那些边e_p^i($p=1$，2，…，k)中原来的两段边代以一条边就行了。如图10—20(d)和(e)所示那样。所以，由$M(i)_1$或$M(i)_2$中一行所代表的任何一个关联组，除行i以外，仍将留下来以表示一个关联组。这样，我们有下一定理。

定理10—11：设一个矩阵C_f的一对M-子矩阵为$M(j_1\cdots j_k i)$和$M(j_1'\cdots j_m' i)$，其中$j_1\neq\cdots\neq j_k\neq j_1'\neq\cdots\neq j_m'\neq i$(注意只有$i$是在$M(j_1\cdots j_k i)$和$M(j_1'\cdots j_m' i)$两者之中)。当且仅当$M(j_1\cdots j_k i)$和$M(j_1'\cdots j_m' i)$可以实现为线性图$G'$和$G''$的基本截割组矩阵，且(1)$G'$中由行$j_1$、…，$j_k$和$i$所代表的那些基本截割组是些关联组，和(2)$G''$中由行$j_1'\cdots j_m'$和$i$所代表的那些基本截割组是关联组时，$C_f$才可以实现为一个线性图的基本截割组矩阵，其中由行$j_1$，…，$j_k$，$j_1'$，…，$j_m'$所代表的那些基本截割组是些关联组。

设$M(i)_1$是一个基本截割组矩阵中关于行i的一对M-子矩阵中的一个，并且$i_2(\neq i_1)$是$M(i)_1$中的一行。让我们考虑$M(i)_1$中关于行i_2的一对M-子矩阵。$M(i)_1$中关于行i_2的这两个M-子矩阵有这样一些特性，就是这两个中的一个同时包括行i_1和行i_2，而另一个只包括行i_2，但是不包括行i_1。同时包括行i_1和行i_2的那个用$M(i_1\ i_2)$来表明，而另一个只包括行i_2而不包括行i_1的用$M(i_2)$来表明。

一般地讲，我们可以得到$M(i_1\ i_2\cdots i_k)$中关于行i_{k+1}而不是行i_1，i_2，…，i_k中某一个的一对M-子矩阵，设这两个子矩阵中的一个包括行i_1，i_2，…，$i_m(m\leqslant k)$，当然也包括行i_{k+1}。我们用符号$M(i_1 i_2\cdots i_m i_{k+1})$来表明这个子矩阵。另一个子矩阵很清楚是$M(i_{m+1}\cdots i_k i_{k+1})$。根据定理10—11：当且仅当$M(i\cdots i_m i_{k+1})$和$M(i_{m+1}\cdots i_k i_{k+1})$可以实现为基本截割组矩阵，但是有一个约束，即那些行i_1，i_2，…，i_m，i_{m+1}，…，i_k和i_{k+1}代表些关联组时，矩阵$M(i_1 i_2\cdots i_m\cdots i_k)$才可以实现为一个基本截割组矩阵，它的那些行$i_1$，$i_2$，…，$i_m$，…，$i_k$代表些关联组。

只要p不在括弧之内，我们就可以组成一个M-子矩阵$M(i_1\cdots i_n)$中关于行p的一对M-子矩阵。当一个M-子矩阵$M(i_1 i_2\cdots i_n)$包括行i_1，i_2，…，i_n时，这个M-子矩阵叫做最小M-子矩阵。所以如果一个M-子矩阵$M(i_1 i_2\cdots i_n)$只包括行i_1，i_2，…，i_n时。这个M-子矩阵就是一个最小M-子矩阵。

因为在一个最小M-子矩阵中，每一行都在括弧之内，每一行一定代表着一个关联组。因此一个最小M-子矩阵一定是一个缩减关联矩阵。

因为每一行我们都可以组成一对M-子矩阵，在一个有n_v个点的一个连通图的一个基本截割组矩阵C_f中，我们可以得到n_v个最小M-子矩阵。这些n_v个最小M-子矩阵组成一个矩阵

C_f的最小M-子矩阵组。

定理10—12：对一个基本截割组矩阵C_f讲，有C_f的一个最小M-子矩阵组，其中每一个最小M-子矩阵是一个缩减关联矩阵。

当一个H-子矩阵可以分块成

$$H=\begin{bmatrix} H_1 & & & \\ & H_2 & & 0 \\ & & \ddots & \\ 0 & & & H_p \end{bmatrix} \tag{10—20}$$

时，就有很多方法来组成M-子矩阵对，因此一个基本截割组矩阵C_f的最小M-子矩阵对不可能是唯一的。这就有下一定理。

定理10—13：当且仅当一个矩阵$R=[R_c \mid U]$有一个最小M-子矩阵组，且组中每一个M-子矩阵是一个缩减关联矩阵时，矩阵R是可以实现为一个线性图的一个基本截割组矩阵。

测试一个矩阵是否是一个缩减关联矩阵是很容易的，因为一个缩减关联矩阵的每一列一定最多有两个1。如果一个矩阵的每列最多有两个1，我们就能够构成一个线性图，使这个矩阵是这个图的一个缩减关联矩阵。

例10—11　假设一个矩阵R为

$$R=\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{array}{ccccc:cccc} a & b & c & d & e & f & g & h & l \end{array} \\ \left[\begin{array}{ccccc:cccc} 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right] \end{array}$$

删除行1中有1的各列，并且再删除行1，得到一个H-子矩阵$H(1)$，它可以分块成

$$H(1)=\begin{array}{c} \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{array}{c:cccc} g & d & e & h & l \end{array} \\ \left[\begin{array}{c:cccc} 1 & 0 & 0 & 0 & 0 \\ \hdashline 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{array}\right] \end{array}$$

所以关于行1的一对M-子矩阵为

$$M(1)_1=\begin{array}{c} \\ 1 \\ 2 \end{array}\begin{array}{c} \begin{array}{ccccc} a & b & c & f & g \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array}$$

$$M(1)_2=\begin{array}{c} \\ 1 \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{array}{cccccccc} a & b & c & d & e & f & h & l \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

$M(1)_1$的H-子矩阵$H(2)$为

$$H(2)=1\begin{array}{c} \begin{array}{cc} b & f \end{array} \\ \begin{bmatrix} 1 & 1 \end{bmatrix} \end{array}$$

因此，取$H(2)_1=1[1\ 1]$和$H(2)_2=\phi$，M-子矩阵$M(12)$可以得自$M(1)$而不删除任何行和列，即

$$M(12)=M(1)_1$$

这是一个最小M-子矩阵。M-子矩阵$M(2)$可以得自$M(1)$而不删除行1和列b和f，即

$$M(2)=\begin{array}{c} \\ 2 \end{array}\begin{array}{c} \begin{array}{ccc} a & c & g \end{array} \\ \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \end{array}$$

这是一个最小M-子矩阵。$M(12)=M(1)_1$表示矩阵$M(12)$中的每个元素值等于矩阵$M(1)_1$中对应的元素值。可是符号$M(12)$和$M(1)_1$却有着不同的约束。这就是说，$M(12)$的行1和行2一定各代表着一个关联组，不过在$M(1)$中只需要行1代表着一个关联组。

$M(1)_2$的H-子矩阵$H(3)$为

$$H(3)=\begin{array}{c} \\ 1 \\ 4 \end{array}\begin{array}{c} \begin{array}{cccc} a & b & f & l \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \end{array}$$

这个矩阵不能再像（10—19）那样分块了，所以我们取$H(3)_2=\phi$，这样有

$$M(13)=M(1)_2$$

和

$$M(3)=\begin{array}{c} \\ 3 \end{array}\begin{array}{c} \begin{array}{cccc} c & d & e & h \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

这是一个M-子矩阵。最后，从$M(13)$中删除在行4中有一的各列并删除行4，我们得到$M(13)$的H-子矩阵$H(4)$并加以分块得

$$H(4)=\begin{array}{c} \\ 1 \\ \\ 3 \end{array}\begin{array}{c} \begin{array}{cc|cc} a & f & d & h \end{array} \\ \left[\begin{array}{cc|cc} 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 \end{array}\right] \end{array}$$

$M(13)$中关于行4的一对M-子矩阵为

$$M(14)=\begin{array}{c} \\ 1 \\ 4 \end{array}\begin{array}{c} \begin{array}{cccccc} a & b & c & e & f & l \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \end{array}$$

这是一个最小的M-子矩阵

$$M(34)=\begin{array}{c} \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{array}{cccccc} b & c & d & e & h & l \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array}$$

这也是一个最小M-子矩阵。既然最小M-子矩阵组$M(12)$，$M(2)$，$M(3)$，$M(14)$和$M(34)$满足定理10—13，所以给出的矩阵R是一个线性图的一个基本截割组矩阵。

从一个图的基本截割组矩阵C_f出发，通过下列步骤可以构成一个线性图。

（一）有了一个线性图的基本截割组矩阵$M(13)$后，先用它的缩减关联矩阵$M(14)$和$M(34)$构成两个图，如图10—21(a)和(b)所示。然后合并而成为一个线性图，如图(c)所示。

（二）将缩减关联矩阵$M(12)$构成一个线性图，如图10—22所示。将这个图与第一步中所得到的图10—21(c)合并即得基本截割组矩阵C_f的线性图，如图10—23所示。

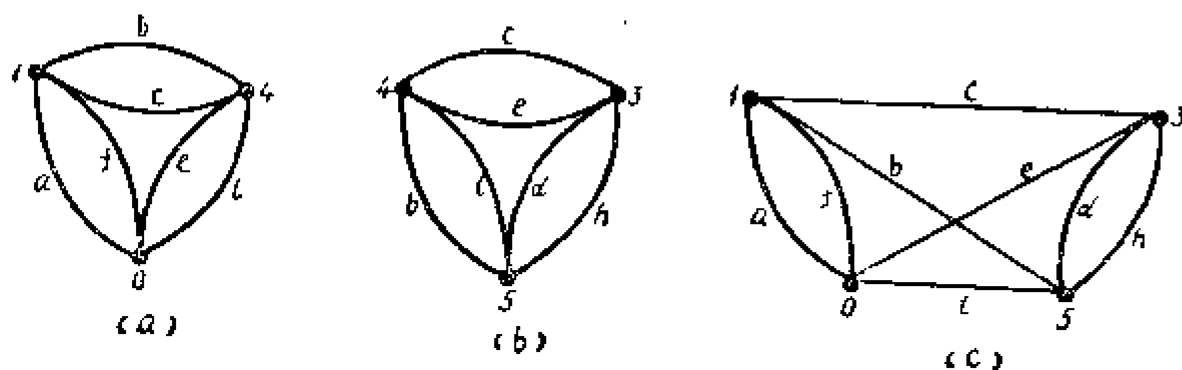


图 10—21

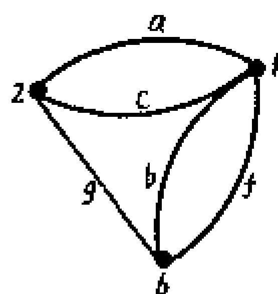


图 10—22

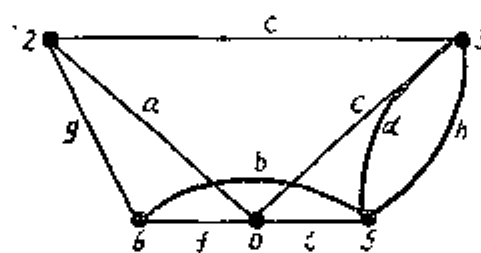


图 10—23

注意，用这些步骤来求取C_f的最后线性图时是用不着只有一行的缩减关联矩阵的任何线性图的。另外，定理10—13也适用于分离图的。

有很多情况，在得到最小M-子矩阵之前，我们就可以发现一个给定的矩阵是不能实现为一个基本截割组矩阵的。下面二定理就给出这样的情况。

定理10—14：设$M(i_1i_2\cdots i_p)$是一个M-子矩阵，如果有一列中在i_1，i_2，…，i_p中至少有3个1，那么$M(i_1i_2\cdots i_p)$就不能实现为行i_1，i_2，…，i_p代表着关联组的一个基本截割组矩阵。

定理10—15：如果包括行i_1，i_2，…，i_p的矩阵$M(i_1i_2\cdots i_p)$的子矩阵不是一个缩减关联矩阵时，那么$M(i_1i_2\cdots i_p)$不能实现为i_1，i_2，…，i_p代表着关联组的一个基本截割组矩阵。

例10—12　有一个矩阵R为

$$R=\begin{array}{c} \\ 1\\ 2\\ 3\\ 4 \end{array}\begin{array}{c} \begin{array}{cccccccc} a & b & c & d & e & f & g & h \end{array} \\ \left[\begin{array}{cccccccc} 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0\\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0\\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0\\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right] \end{array}$$

取行1，H-子矩阵$H(1)$为

$$H(1)=\begin{array}{c} \\ 2\\ 3\\ 4 \end{array}\begin{array}{c} \begin{array}{cccc} b & f & g & h \end{array} \\ \left[\begin{array}{cccc} 1 & 1 & 0 & 0\\ 1 & 0 & 1 & 0\\ 1 & 0 & 0 & 1 \end{array}\right] \end{array}$$

这个矩阵不能分块，我们取$H(1)_2=\phi$。所以M-子矩阵$M(1)_1$和$M(1)_2$为

$$M(1)_1=R$$

$$M(1)_2=1\begin{array}{c} \begin{array}{cccc} a & c & d & e \end{array} \\ \left[\begin{array}{cccc} 1 & 1 & 1 & 1 \end{array}\right] \end{array}$$

取$M(1)_1$的行 2，它的H-子矩阵$H(2)$为

$$H(2)=\begin{matrix} & c & e & g & h \\ 1 & 1 & 1 & 0 & 0 \\ 3 & 1 & 0 & 1 & 0 \\ 4 & 1 & 0 & 0 & 1 \end{matrix}$$

所以$M(1)_1$中关于行 2 的一对M-子矩阵为

$$M(12)_1=R$$

$$M(1)_1=\begin{matrix} & a & b & d & f \\ 2 & 1 & 1 & 1 & 1 \end{matrix}$$

同样，$M(12)$中关于行 3 的一对M-子矩阵为

$$M(123)_1=R$$

$$M(3)_1=\begin{matrix} & a & b & c & g \\ 3 & 1 & 1 & 1 & 1 \end{matrix}$$

因$M(123)_1$中包含下列的子矩阵

$$\begin{matrix} & a & b & c & d & e & f & g \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 2 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 3 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{matrix}$$

其中每行是在括弧之中（就是每行一定代表着一个关联组），这个子矩阵一定是一个缩减关联矩阵。可是，列a有 3 个 1，所以这个子矩阵不可能是一个缩减关联矩阵。因此根据定理10—15，在这样的M-子矩阵组的情况下，R不可能实现为一个基本截割组矩阵。因为用行 1，2和 3 不能得到其它的M-子矩阵组，所以我们可以判定这个矩阵R不是一个基本截割组矩阵。

为了去除不可实现的矩阵，下面的定理是一条重要定理。

定理10—16：对于一个给定的矩阵$R=[R_c \mid U]$，如果有一个子矩阵$R'=[R'_c \mid U']$是不能实现为一个基本截割组矩阵时，那么矩阵R是不可实现为一个基本截割组矩阵时，那么矩阵R是不可实现为一个基本截割组矩阵的，其中U'是一个单位矩阵。

证：假使$R=[R_c \mid U]$是一个关于生成树T的一个基本截割组矩阵。因为在矩阵中删除一些行相当于在T中短捷一些枝。我们可以删除一些行使余留下来的矩阵R_a的行数与R'的行数正相等。注意$R_a=[R_{ac} \mid U]$是一个线性图的一个基本截割组矩阵。可以看出删除R_a的一些列等于删除掉对应这些列的一些弦。这样我们删除掉只在R_{ac}中而不在R'_c中的一些列而能够得到一个线性图的一个基本截割组矩阵R'的。这是与R'不能实现性的假定相违背的。所以R不应是一个基本截割组矩阵。▌

例10—13　因为矩阵

$$L=\begin{array}{c} \begin{matrix} a & b & c & d & e & f & g \end{matrix} \\ \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

不能实现为一个基本截割组矩阵（见例10—12）。如果任何一个矩阵R的形式为$[R_c \mid U]$，并包含L矩阵作为它的子矩阵时，R是不能实现为一个基本截割组矩阵的。

上面讲的测试一个截割组矩阵成为一个图的可实现性时，我们需要对这一个给定的矩阵加以运算。这个算法对于在某些限制条件下组成一个基本截割组矩阵也许是不方便的。

现在我们先研究一下一个平面图中一个截割组矩阵与一个回路矩阵间的关系。

假设$[U \mid B_t]$是一个平面图G_1的一个基本回路矩阵。根据定理6—20，矩阵$[B_t \mid U]$是G_1的对偶图G_2的基本截割组矩阵。同样，如果$[C_c \mid U]$是一个平面图G_1的一个截割组矩阵，那么矩阵$[U \mid C_c]$是G_1的对偶图G_2的一个回路矩阵。另一方面，假设矩阵$[N \mid U]$是一个非平面图的一个回路矩阵，那么根据定理6—21，一个非平面图不可能有对偶，所以$[N \mid U]$不可能是一个基本截割组。另外，定理10—16说如果一个矩阵$[N \mid U]$含有一个不是截割组矩阵的子矩阵，那么这个矩阵就不是一个基本截割组矩阵。

定理10—17：如果一个矩阵$[N \mid U]$有一个子矩阵是两个典型的非平面图中之一或它的同胚的一个基本回路矩阵，那么这个矩阵就不是一个基本截割组矩阵。

下列的矩阵，用符号N_0表示，

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

有一个特殊的性质，就是凡是有N_0或N_0'组成的矩阵，如$[N_0 \mid U_0]$或$[N_0' \mid U_0']$，都不可能是一个基本截割组矩阵，其中U_0和U_0'都是单位矩阵。有了这个特性，上面的定理可以扩充为下面这一定理。

定理10—18：如果一个矩阵$[N \mid U]$含有(1)$[N_0 \mid U_0]$，(2)$[N_0' \mid U_0']$，和(3)两个典型的非平面图之一或它的同胚的一个基本回路矩阵，这三者中之一的，这个矩阵就不是一个基本截割组矩阵。

定理10—19：当且仅当一个矩阵$[N \mid U]$不含有下列之一时：(1)$[N_0 \mid U_0]$，(2)$[N_0' \mid U_0']$，和(3)两个典型的非平面图之一或它的同胚的一个基本回路矩阵，这个矩阵才是一个基本截割组矩阵。

根据式（5—82）或式（5—83）所给出的一个截割组矩阵与一个回路矩阵的关系，上一定理也可修改为

定理10—20：当且仅当一个矩阵$[N \mid U]$不含有下列之一时，(1)$[N_0 \mid U_0]$，(2)$[N_0' \mid U_0']$，和(3)两个典型的非平面图中之一或它的同胚的一个基本截割组矩阵，这个矩阵才是一个基本回路矩阵。

对于无向图，我们采用M-子矩阵来测试一个矩阵是否是一个基本截割组矩阵。对于有向图呢，我们也可以用相同的方法来测试一个基本截割组矩阵。可是我们一定要先研究一个有向图的一个基本截割组矩阵的特性。

定理10—21：如果一个矩阵$C_f=[C_c \mid U]$是一个有向图G的一个基本截割组矩阵，那么C_f也是另一有向图G'，它是把G中各边的定向颠倒过来而得到的有向图的一个基本截割组矩阵。

证：当一个图的各边的定向颠倒过来后，我们得到相同的截割组矩阵，只是那些截割组的定向相反而已。▌

下一定理表示一对M-子矩阵是重要特性，它导致一个有向图的一个基本截割组矩阵的可实现性。

定理10—22：设一对M-子矩阵$M_{1+2}=[M_e \vdots U]$关于行p为M_1和M_2。假设有一有向图$g_k(k=1, 2)$，使(1)g_k的基本截割组矩阵$M_k(k=1, 2)$，(2)在g_k中有一个点p，使M_k中的行p或M_k中(-1)乘行p代表关于点p的一个缩减关联矩阵，$k=1, 2$。那么就有一个图g_{1+2}，使(1)M_{1+2}是g_{1+2}的一个基本截割组矩阵，和(2)对于M_k中除行p以外的其它每一行q，如果不是行q就是(-1)乘行q代表q_k的一个缩减关联矩阵，那么M_{1+2}中或是行q或是(-1)乘行q就代表着g_{1+2}中的一个缩减关联矩阵，$k=1, 2$。

这个定理与定理10—13相似。

定理10—23：当且仅当一个矩阵$R=[R_e \vdots U]$有一组最小M-子矩阵，组中每一个最小M-子矩阵的某些行乘以(-1)时就成为一个缩减关联矩阵时，这个矩阵R是一个有向图的一个基本截割组矩阵。

注意，当且仅当一个矩阵包括$+1$，-1和0，而且每一列有着一个非零值或正负号相反的两个非零值时，这个矩阵是一个缩减关联矩阵。我们知道，对于一个有向图。关于点v的一个缩减关联矩阵可以表达为$E(V\times\overline{V})\cup E(\overline{V}\times V)$，它的定向是从$V$到$\overline{V}$的。我们也知道对于一个点$p$，$E(p\times\overline{p})\cup E(\overline{p}\times p)$或是一个截割组或是截割组的边不共用并集。假设$E(p\times\overline{p})\cup E(\overline{p}\times p)$是从$\overline{p}$到$p$，那么因为定向的相反，组$E(p\times\overline{p})\cup E(\overline{p}\times p)$不是一个缩减关联矩阵。假设这个组由一个截割组矩阵的行p所代表。我们知道(-1)乘行p等于把这组的定向倒过来。所以(-1)乘行p后，代表行p的组$E(p\times\overline{p})\cup E(\overline{p}\times p)$就变成一个缩减关联矩阵。有了这个事实，再应用定理10—22，我们就可以证明定理10—23了。

例10—14　有一个矩阵R为

$$R=\begin{array}{c}\\1\\2\\3\\4\end{array}\begin{array}{cccccc:cccc} a & b & c & d & e & & f & g & h & l\\ -1 & 1 & 1 & 0 & 0 & & 1 & 0 & 0 & 0\\ 1 & 0 & -1 & 0 & 0 & & 0 & 1 & 0 & 0\\ 0 & 0 & 1 & -1 & -1 & & 0 & 0 & 1 & 0\\ 0 & 1 & 1 & 0 & -1 & & 0 & 0 & 0 & 1\end{array}$$

它的一组最小M-子矩阵为

$$M_1=\begin{array}{c}\\1\\2\end{array}\begin{array}{ccccc} a & b & c & f & g\\ -1 & 1 & 1 & 1 & 0\\ 1 & 0 & -1 & 0 & 1\end{array}$$

$$M_2=\begin{array}{c}\\2\end{array}\begin{array}{ccc} a & c & g\\ 1 & -1 & 1\end{array}$$

$$M_3=\begin{array}{c}\\3\end{array}\begin{array}{cccc} c & d & e & h\\ 1 & -1 & -1 & 1\end{array}$$

$$M_4=\begin{array}{c}\\1\\4\end{array}\begin{array}{cccccc} a & b & c & e & f & l\\ -1 & 1 & 1 & 0 & 1 & 0\\ 0 & 1 & 1 & -1 & 0 & 1\end{array}$$

$$M_5=\begin{array}{c} \\ 3 \\ 4 \end{array}\begin{array}{c}\begin{array}{cccccc} b & c & d & e & h & l \end{array}\\ \left(\begin{array}{cccccc} 0 & 1 & -1 & -1 & 1 & 0 \\ 1 & 1 & 0 & -1 & 0 & 1 \end{array}\right)\end{array}$$

将M_4的行 4 和M_5的行 4 都乘以（—1）得

$$M_4^1=\begin{array}{c} \\ 1 \\ 4 \end{array}\begin{array}{c}\begin{array}{cccccc} a & b & c & e & f & l \end{array}\\ \left(\begin{array}{cccccc} -1 & 1 & 1 & 0 & 1 & 0 \\ 0 & -1 & -1 & 1 & 0 & 1 \end{array}\right)\end{array}$$

$$M_5^1=\begin{array}{c} \\ 3 \\ 4 \end{array}\begin{array}{c}\begin{array}{cccccc} b & c & d & e & h & l \end{array}\\ \left(\begin{array}{cccccc} 0 & 1 & -1 & -1 & 1 & 0 \\ -1 & -1 & 0 & 1 & 0 & -1 \end{array}\right)\end{array}$$

因为M_1，M_2，M_3，M_4^1和M_5^1都是缩减关联矩阵，根据定理10--23，矩阵R是一个有向图的缩减关联矩阵。

第五节　开关网络的合成

所谓网络的合成就是根据给定的要求，设计一个网络。在开关网络或接点网络的合成中，要求都是用开关函数的方式给出的。如果用其它方式给出时，它们也是可以转变成开关函数的。

一个开关函数可以表达成各种不同方式，像在本章第二节中已经讲到过的标准和式和标准乘积式。要求合成网络时所给出的开关函数，我们假定是用标准和式表达的。

在合成一个二端开关网络时，给出一个标准和式的有m个变数的开关函数 $F_{ij}(x_1,x_2,\cdots,x_m)$，要求设计出一个网络在点 i 和 j 之间实现函数所示的传输情况。

如果我们不考虑节省接点数目的问题，那么合成网络是很容易的。因为用大量的接点总是可以实现一个开关函数的。可是这种浪费现象一般是不能接受的。一个好的网络设计必须要用越少越好的接点数来实现开关函数的要求。由于这个要求，合成就不是一件轻易的事情了。图10—24中有 3 个图，都能实现一个很简单的开关函数 $F_{12}=wx+wy+wz+xyz$（只有 4 个变数）的要求。这 3 个网络中最简单的一个是图（c）。我们也许要问，这一个是否是可能实现方法中最经济的一个呢？如果是的，我们怎能确信呢？对某一开关函数，是否有任何方法能保证我们获得一个最经济的接点网络呢？

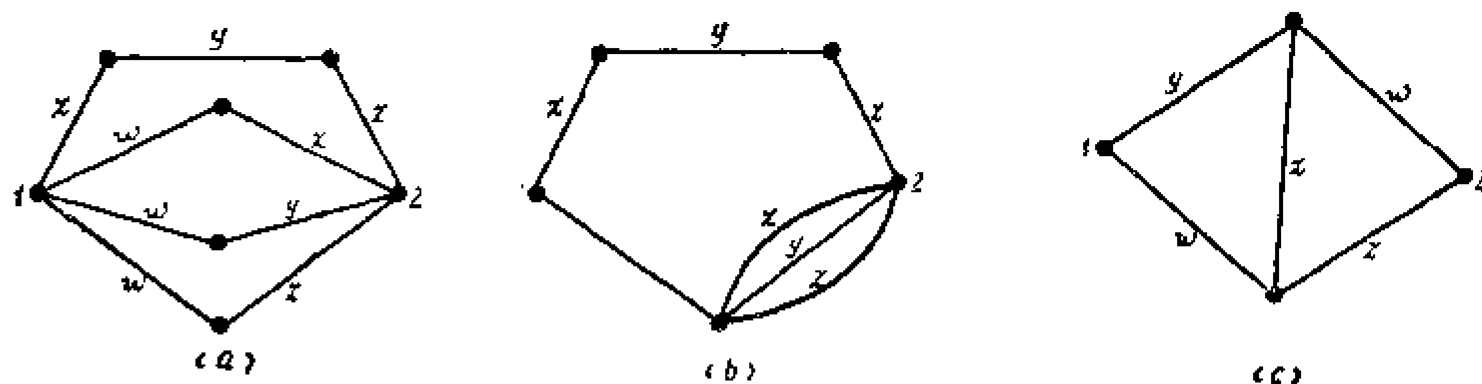


图　10—24

用最少接点数来实现一个任意开关函数至今还没有解决（除非彻底地列举各种方案），而且看起来，在最近的将来也不会解决的。可是，如果我们只考虑开关网络中个别方式叫作单接点网络，或SC网络，那么问题就容易处理了。

所谓单接点网络是一种无向图，其中每一条边有不同的布尔变数（或系变数，或系它的补数），这样在单接点网络中，每个接点可以单独地或不相关地开放或闭合。例如，图10—6是一个单接点网络，而图10—24则不是。

如果一个开关函数能够用一个单接点网络来实现时，这个函数就叫作单接点函数或SC函数。既然在一个单接点函数中一个变数只出现一次，所以这个表达式是无法简化的。换句话说，一个单接点函数是不含有冗余项的。每一个乘积项代表着某规定两点之间的一条不同的路径，并且每一乘积项中每一个文字对应着路径中一条不同的边。例如，$F=ab+acd+dc$不是一个单接点函数，而$F=ab+acd$是一个单接点函数。

在这一节中，我们只考虑那些开关函数是完全确定的，所谓完全确定的函数是指它的真值表是1或是0，而不是没有确定的。

如果集体$\{P_{rs}\}$中所有元素P_{rs}的乘积与开关函数F_{rs}有下列关系

$$F_{rs}=\sum_{P_{rs}\in\{P_{rs}\}}P_{rs}\text{乘积} \qquad (10\text{—}21)$$

那么$\{P_{rs}\}$叫作对应一个完全确定了的开关函数F_{rs}的一个集体。

例如，如果$F_{rs}=a(b+ce)+d(e+bc)$，那么对应着F_{rs}的集体$\{P_{rs}\}$为

$$\{P_{rs}\}=\{(a,b),\ (a,c,e),\ (d,e),\ (b,c,d)\}$$

因为在一个单接点网络中每边的权是不同的，我们就方便地以一边的权作为一边的名称。当我们把一个开关函数表达为

$$F_{ij}=\sum_{P_{ij}\in\{P_{ij}\}}P_{ij}\text{乘积}$$

时，我们可以看出如果对应于F_{ij}的开关网络是一单接点网络的话，$\{P_{ij}\}$一定是点i和j之间所有可能路径的集体。由于这个理由，一个单接点函数应有下列必要条件。

定理10—24，对于$\{P_{ij}\}$中P_r的任何奇数。比方说P_1，P_2，…，P_{2k+1}，其中集体$\{P_{ij}\}$对应着一个单接点网络的完全确定的开关函数F_{ij}，那么在$\{P_{ij}\}$中有P'

$$P'\subset P_1\oplus P_2\oplus\cdots\oplus P_{2k+1} \qquad (10\text{—}22)$$

证：如果$\{P_{ij}\}$是点i和j之间所有可能路径的集体，那么奇数$P_r\in\{P_{ij}\}$的环和是一个M—图的M_{ij}，它或是一个路径P，或是一个路径P和一些回路的一个边不共用并集，其中$P\in\{P_{ij}\}$。

例如，如果$F_{ij}=a(b+ce)+bcd$。那么$\{P_{ij}\}=\{(a,b),\ (a,c,e),\ (b,c,d)\}$。因为$(a,b)\oplus(a,c,e)\oplus(b,c,d)=(d,e)$，而且没有$P'\in\{P_{ij}\}$可以满足式（10—25），所以$F_{ij}$不是一个单接点网络的完全确定的开关函数。

现在让我们复习一些对单接点网络的合成有帮助的子图集体的特性。在上一节中，我们已经知道如何从一个基本截割组矩阵C_f来合或一个线性图。从第五章第八节中我们有在一个基本截割组矩阵C_f与一个基本回路矩阵之间的关系如下

$$C_f=[B\,|\,U_{n_{v}-1}] \qquad (10\text{—}23)$$

从定理2—7中我们知道一个基本回路矩阵是很容易从所有可能回路集体$\{C\}$中得到的，即

$$\{C\}=\min\{E\} \qquad (10\text{—}24)$$

其中$\{E\}$是一个线性图G中所有可能欧拉图的集体。所以如果给出一个集体$\{E\}$，我们就可

以得到一个线性图G。

一个单接点函数F_{ij}将给出点 i 和 j 之间的所有可能路径集体$\{P_{ij}\}$，但不是$\{E\}$。可是从第二章第十一节中，我们知道$\{E_{ij}M_{ij}\}$有n_e-n_v+2个生成元，其中n_e和n_v分别是一个连通图的边数和点数。既然我们要从一个完全确定了的单接点函数F_{ij}来合成一个单接点网络G，我们可以假定这个网络或图G是连通的。而且，从F_{ij}合成的单接点网络G有一特性，就是每一条边至少是在从 i 到 j 的路径中。因此，可以看出G或是不可分的，或是包括k个不可分子图g_1，g_2，…，g_k，它们是（1）这些子图中没有两个是有共同边的，（2）这些子图中任何两个至多有一个共同点，（3）有$k-1$个截割点，（4）$i\in g_1$和$j\in g_k$和（5）从 i 到 j 的每一条路径经过所有$k-1$个截割点，如图10—25所示。这样，可以看出

$$\{P_{ij}\}\otimes\{P_{ij}\}=\{C\} \tag{10—25}$$

其中$\{C\}$是G中所有回路集体。

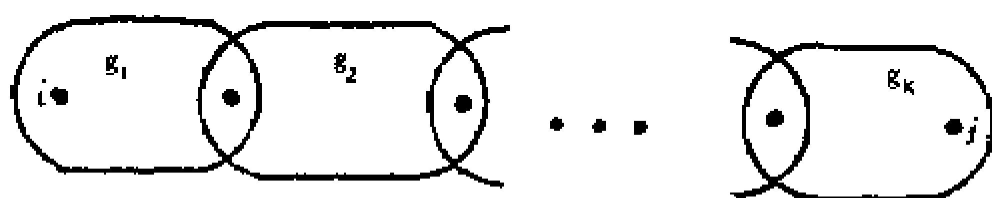


图 10—25

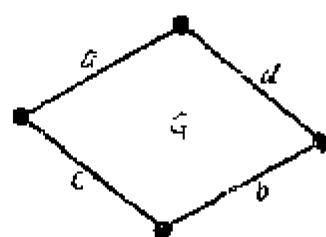


图 10—26

从$\{C\}$中我们可以得到一个基本回路矩阵B_f。由式（10—23）我们能够得到一个基本截割组矩阵C_f。现在我们可以找到一个线性图，它有$\{C\}$作为所有可能回路的一个集体。可是，G不一定有$\{P_{ij}\}$作为从 i 到 j 的所有可能路径的一个集体。例如，假使$\{P_{ij}\}=\{(a,b),\ (c,d)\}$，那么

$$\{P_{ij}\}\otimes\{P_{ij}\}=\{\phi,(a,b,c,d)\}$$

图10—26中的G有(a,b,c,d)作为一个回路。因此G有$\{C\}$作为一个所有可能回路的集体。可是，G没有$\{(a,b),\ (c,d)\}$作为两点之间的所有可能路径的集体。

为了得到一个线性图，它有$\{P_{ij}\}$作为点 i 和 j 之间的所有可能路径集体。我们插入一条边y，以点 i 和 j 作为这条边的端点。换句话说，如果有一个单接点网络G，它有$\{P_{ij}\}$作为点 i 和 j 之间的所有可能路径集体，那么插入一条边y后，一个新的线性图$G\cup(y)$将有一个特性，就是，对每个$P\in\{P_{ij}\}$讲，$(y)\cup P$是一个回路。因此

$$\{(y)\cup P_{ij}\}=\{(y)\cup P\mid P\in\{P_{ij}\}\} \tag{10—26}$$

是$G\cup(y)$中所有可能回路集体$\{C\}$中的一个子组。

既然我们的单接点网络有一个特性

$$\{P_{ij}\}\otimes\{P_{ij}\}=\{C\}$$

$\{(y)\cup P_{ij}\}$将包括所有线性无关回路，它们能产生$G\cup(y)$中的所有回路。因此，利用$\{(y)\cup P_{ij}\}$中的回路。我们能够得到$G\cup(y)$中的一个回路矩阵B_y。从这个回路矩阵B_y出发，我们可以得到一个基本回路矩阵B_f和一个基本截割组矩阵C_f。有了这个矩阵C_f，我们就可以构成一个线性图G'了。

现在一个重要的疑问是从G'中删除边y后，能否给出一个单接点网络，它含有给出的$\{P_{ij}\}$作为两点之间的一个所有可能路径的集体。下面的定理回答了这个疑问。

定理10—25：设$\{P_{ij}\}$是对应着一个完全确定了的开关函数F_{ij}，它满足定理10—24。设P_1，P_2，…，P_m是$\{P_{ij}\}$中线性独立的路径，且$\{P_{ij}\}$中任何一个其它路径P'与这些路径不能是线性独立的。当且仅当有一个圈数为N的无向图G，它的线性独立回路是$(y)\cup P_1$，(y)

$\cup P_2$, …, $(y)\cup P_N$，其中y不在$\{P_{ij}\}$的任何一条路径之中时，没有边y的G是一个满足F_{ij}的单接点网络。

证：假设有一个圈数为N的无向图G，其中$(y)\cup P_1$，$(y)\cup P_2$，…，$(y)\cup P_N$是些线性独立的回路。假设在G的点 i 和 j 之间有一条路径P'，它不在$\{P_{ij}\}$之中。既然$(y)\cup P'$是G中的一个回路，$(y)\cup P'$可以用$(y)\cup P_1$，$(y)\cup P_2$，…，$(y)\cup P_N$中几个的环和得到的。再推而广之，设

$$(y)\cup P'=\{(y)\cup P_1\}\oplus\{(y)\cup P_2\}\oplus\cdots\oplus\{(y)\cup P_{2k+1}\} \qquad (10-27)$$

其中$2k+1\leqslant N$，那么

$$P'=P_1\oplus P_2\oplus\cdots\oplus P_{2k+1} \qquad (10-28)$$

因为F_{ij}满足定理10—24，P'一定是在$\{P_{ij}\}$之中。但是我们上面假设P'不在$\{P_{ij}\}$之中，所以这是矛盾的。

假使没有边y的G是一个满足F_{ij}的单接点网络，那么很明显，G的圈数为N，并且$(y)\cup P_1$，$(y)\cup P_2$，…，$(y)\cup P_N$是线性独立的回路。

注意这个定理中所谓的"满足F_{ij}的单接点网络"的意思是说，在单接点网络中两个点，在这两点之间的开关函数是F_{ij}。这里的"路径"是用来作为$\{P_{ij}\}$中的一个元素的。如果没有一个单接点网络能满足F_{ij}时，这个词就不确切了。

应用这条定理，我们可以从一个完全确定了的开关函数F_{ij}依照下列步骤合成一个单接点网络。

（一）测试给出的函数F_{ij}是否满足定理10—24。如果不能的话，F_{ij}不是一个单接点函数。

（二）从对应F_{ij}的集体$\{P_{ij}\}$中取线性独立路径P_1，P_2，…，P_N。用回路$(y)\cup P_1$，$(y)\cup P_2$，…，$(y)\cup P_N$组成一个回路矩阵B。

（三）依靠行和列的初等运算，改变B为一个基本回路矩阵$B_f=[U\,|\,B_{f}]$。

（四）应用式（10—23），从B_f得到一个基本截割组矩阵$C_f=[B\,|\,U]$。

（五）用上一节中的方法从C_f得到一线性图G'。

（六）删除边y，结果的图G是一个满足F_{ij}的单接点网络，其中 i 和 j 是边y的端点。

例10—15　合成下列完全确定了的开关函数

$$F_{ij}=ab+acd+bce+de$$

（一）取这个开关函数的前 3 项进行环和

$$P'=\{(a,b)\oplus(a,c,d)\oplus(b,c,e)\}=(d,e)$$

这个$P'\in\{P_{ij}\}$之中。因为这个开关函数能满足定理10—24，所以它是一个单接点函数。

（二）从$\{P_{ij}\}=\{(a,b)$，(a,c,d)，(b,c,e)，$(d,e)\}$中，假使我们取$P_1=(a,b)$，$P_2=(a,c,d)$，和$P_3=(b,c,e)$作为线性独立路径。由第一步知$P_1\oplus P_2\oplus P_3=(d,e)$。一个回路矩阵$B$为

$$B=\begin{matrix}(y)\cup P_1\\(y)\cup P_2\\(y)\cup P_3\end{matrix}\overset{\begin{matrix}a&b&c&d&e&y\end{matrix}}{\begin{pmatrix}1&1&0&0&0&1\\1&0&1&1&0&1\\0&1&1&0&1&1\end{pmatrix}}$$

（三）将B中的一些列进行置换，得

$$B = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{cccccc} b & d & e & a & c & y \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \end{array}$$

将行1加到行3，得

$$B_f = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \left(\begin{array}{ccc|ccc} b & d & e & a & c & y \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{array}\right) = [U \vdots B_t]$$

（四）根据式（10—23），由B_f得C_f

$$C_f = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \left(\begin{array}{ccc|ccc} b & d & e & a & c & y \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right)$$

（五）我们先得到关于行1的H-子矩阵$H(1)$

$$H_{(1)} = \begin{array}{c} \\ 2 \\ 3 \end{array} \left(\begin{array}{c|c} c & y \\ 1 & 0 \\ \hline 0 & 1 \end{array}\right)$$

有了这个H-子矩阵，我们可以得到一对M-子矩阵$M(1)_1$和$M(1)_2$：

$$M(1)_1 = \begin{array}{c} \\ 1 \\ 2 \end{array} \left(\begin{array}{ccccc} b & d & e & a & c \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{array}\right)$$

$$M(1)_2 = \begin{array}{c} \\ 1 \\ 3 \end{array} \left(\begin{array}{ccccc} b & d & e & a & y \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{array}\right)$$

对应$M(1)_1$和$M(1)_2$的两个线性图g_1和g_2，如图10—27所示。合并g_1和g_2得一线性图G_y如图10—28(a)所示。

（六）从G_y中删除边y即得到所求的单接点网络G，在点v_2和v_3（这是边y原来的端点）之间满足给出的开关函数。

合成单接点网络的方法不止一种，下面是另一种方法，这种方法与前一种方法大同小异。首先也是要测试一个给出的开关函数F_{ij}是否是一个单接点函数。然后按照下列步骤进行。

（一）从给出的开关函数F_{ij}得到一个关于点i和j之间的路径矩阵$P_{(ij)}$。

（二）在矩阵$P_{(ij)}$末增加一列。这就意味着在点i和j之间增加一条边，因此把各条路径变成各个回路。设所得的回路矩阵为B。

（三）在B中用模2消除法消去所有的相依行，并加以置换，以得到基本回路矩阵$B_f=[U \vdots B_t]$。

（四）应用式（10—23），从B_f得到基本截割组矩阵$C_f=[B_t' \vdots U]=[C_t \vdots U]$。

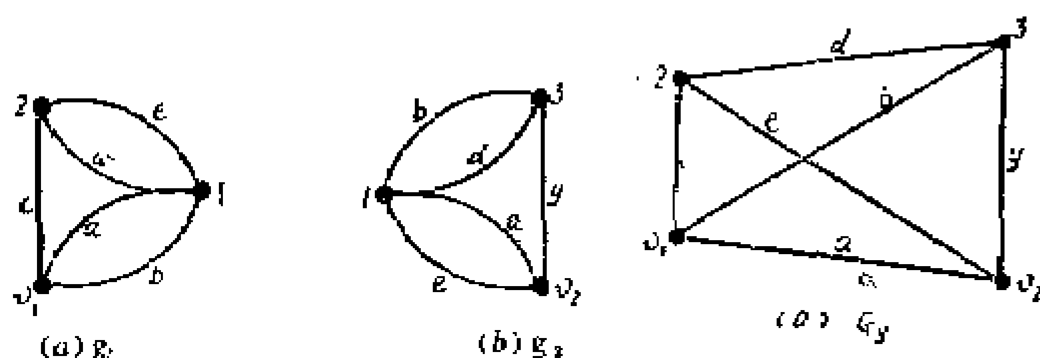


图 10—27

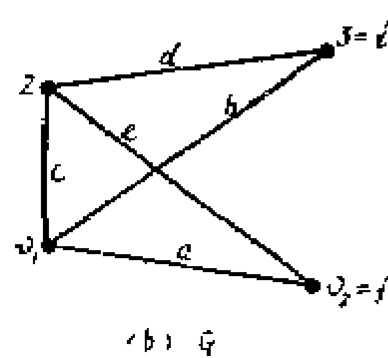


图 10—28

（五）在C_f的各行中进行适当的模 2 加法，以得到缩减关联矩阵A_f。这就相当于得到一个非奇异变换矩阵R，使

$$A_f = RC_f \qquad (10-29)$$

式中A_f的每一行最多只能有两个 1，在第五步中，本质上我们是把各基本截割组进行不同的环和运算，使产生关联到各点的边组。在整个过程中，这一步最为烦琐，并且对于大的线性图，可能进行不下去的。

（六）对A_f补足失去的行使成为关联矩阵A，使每列正好有两个 1。由矩阵A画出线性图，并删除第二步中增加的那一条边，即为所求的单接点网络。

例10—16　合成下列单接点函数

$$F_{ij} = x_1x_2x_3x_5x_7 + x_1x_3x_4x_6 + x_1x_5x_6x_8 + x_2x_4 + x_2x_3x_5x_8 + x_3x_4x_6x_7x_8 + x_5x_6x_7$$

（一）路径矩阵$P_{(ij)}$为

$$P_{(ij)} = \begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \end{matrix} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

（二）在矩阵$P_{(ij)}$的末端增加都是 1 的一列，这列标以x_0，我们得到一个回路矩阵B

$$B = \begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_0 \end{matrix} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix} \end{array}$$

（三）这一步多少有些复杂。所谓消除相依行是把一些行用模 2 加法加到另一些行以便形成一个单位子矩阵。例如，在B中将行 1 加到行 2 和行 3，得

$$B_1=\begin{array}{c} \begin{array}{ccccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_0 \end{array} \\ \left(\begin{array}{ccccccccc} 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{array}\right) \end{array}$$

在B_1中，将行2加到行4和行6，得

$$B_2=\begin{array}{c} \begin{array}{ccccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_0 \end{array} \\ \left(\begin{array}{ccccccccc} 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{array}\right) \end{array}$$

这样运算的目的是为了使一些列中只留一个1，而且也要注意到使每列中的1出现在不同的行中。最后得

$$B_3=\begin{array}{c} \begin{array}{ccccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_0 \end{array} \\ \left(\begin{array}{ccccccccc} 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right) \end{array}$$

从B_3我们得到标准形式的基本回路矩阵B_f

$$B_f=\begin{array}{c} \begin{array}{cccc|ccccc} x_1 & x_4 & x_8 & x_6 & x_2 & x_3 & x_5 & x_7 & x_0 \end{array} \\ \left(\begin{array}{cccc|ccccc} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{array}\right) \end{array}$$

由此我们得到所需网络的一些信息。

回路矩阵B的秩数$R=N=n_e-n_v+1=4$

边数（包括边x_0）$n_e=9$

由此可见　$n_v=n_e-3=6$

截割组矩阵的秩数$=n_v-1=5$

（四）对应第三步中B_f的同一生成树的基本截割组矩阵为

$$C_f=[B_f^t \mid U_{n_v-1}]=\begin{array}{c} \begin{matrix} x_1 & x_4 & x_8 & x_6 & x_2 & x_3 & x_5 & x_7 & x_0 \end{matrix} \\ \left[\begin{array}{cccc:ccccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \end{array}$$

（五）试算多次后我们得到，如果对C_f进行下列三次初等行运算，我们得到一个矩阵，每列最多有两个 1。这些模 2 运算是

行 5 加到行 1

行 3 加到行 5

行 4 加到行 3

所得的缩减关联矩阵A_f为

$$A_f=\begin{array}{c} \begin{matrix} x_1 & x_4 & x_8 & x_6 & x_2 & x_3 & x_5 & x_7 & x_0 \end{matrix} \\ \left[\begin{matrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{matrix}\right] \end{array}$$

（六）为了得到一个关联矩阵A，我们在A_f的底上加一行，使每列正好有两个 1

$$A=\begin{array}{c} \begin{matrix} x_1 & x_4 & x_8 & x_6 & x_2 & x_3 & x_5 & x_7 & x_0 \end{matrix} \\ \left[\begin{matrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{matrix}\right] \end{array}$$

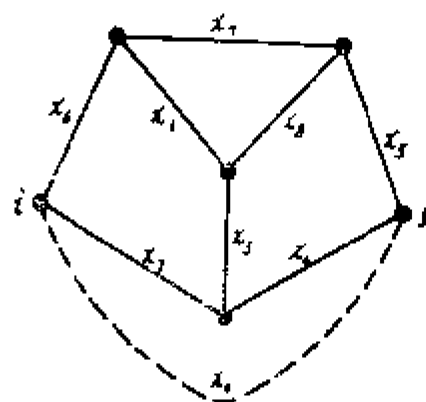


图 10—29

最后根据关联矩阵A构成一图，如图10—29所示，并把边x_0（虚线）删除，就得所需要的单接点网络。

例10—17　合成单线继电半自动闭塞中的正电继电器ZDJ电路。根据ZDJ的励磁条件，其开关函数为

$$F=\overline{a}\,\overline{b}\,c\,\overline{d}\,\overline{e}\,\overline{f}\,h+\overline{a}\,\overline{b}\,\overline{c}\,d\,\overline{e}\,h+\overline{a}\,\overline{b}\,\overline{c}\,\overline{e}\,g$$

式中　a ——正线路继电器ZXJ；

b ——负线路继电器FXJ；

c ——闭塞继电器BSJ；

d ——同意接车继电器TJJ；

e ——回执到达继电器HDJ；

f ——准备开通继电器ZKJ；

g ——开通继电器KTJ；

h·——闭塞按钮。

这个函数的路径矩阵为

$$
\begin{array}{c} \\ P_1 \\ P = P_2 \\ P_3 \end{array}
\begin{array}{c}
\begin{array}{cccccccccc} \bar{a} & \bar{b} & c & \bar{c} & d & \bar{d} & \bar{e} & \bar{f} & g & h \end{array} \\
\left(\begin{array}{cccccccccc}
1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0
\end{array}\right)
\end{array}
$$

构成回路矩阵

$$
\begin{array}{c} \\ C_1 \\ B = C_2 \\ C_3 \end{array}
\begin{array}{c}
\begin{array}{ccccccccccc} \bar{a} & \bar{b} & c & \bar{c} & d & \bar{d} & \bar{e} & \bar{f} & g & h & y \end{array} \\
\left(\begin{array}{ccccccccccc}
1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1
\end{array}\right)
\end{array}
$$

其基本回路矩阵B_f为

$$
\begin{array}{c} \\ C_1 \\ B_f = C_2 \\ C_3 \end{array}
\begin{array}{c}
\begin{array}{ccc|cccccccc} c & d & g & \bar{a} & \bar{b} & \bar{c} & \bar{d} & \bar{e} & \bar{f} & h & y \end{array} \\
\left(\begin{array}{ccc|cccccccc}
1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1
\end{array}\right)
\end{array}
$$

由$C_f = [B_f^T \vdots U]$

$$
C_f = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array}
\begin{array}{c}
\begin{array}{ccc|cccccccc} c & d & g & \bar{a} & \bar{b} & \bar{c} & \bar{d} & \bar{e} & \bar{f} & h & y \end{array} \\
\left(\begin{array}{ccc|cccccccc}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}\right)
\end{array}
$$

把C_f进行模2运算，可得缩减关联矩阵A_f如下：

$$
A_f = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array}
\begin{array}{c}
\begin{array}{ccccccccccc} c & d & g & \bar{a} & \bar{b} & \bar{c} & \bar{d} & \bar{e} & \bar{f} & h & y \end{array} \\
\left(\begin{array}{ccccccccccc}
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{array}\right)
\end{array}
$$

最后可得关联矩阵A如下：

$$
\begin{array}{c}
 \\ \\ \\ \\ A= \\ \\ \\ \\ \\
\end{array}
\begin{array}{c}
 \\ 1\\2\\3\\4\\5\\6\\7\\8\\9
\end{array}
\begin{array}{ccccccccccc}
c & d & g & \bar{a} & \bar{b} & \bar{c} & \bar{d} & \bar{e} & \bar{f} & h & y \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0
\end{array}
$$

由关联矩阵直接画出线性图，如图10—30所示。

在画出线性图后，删除边y，进而考虑到前后接点的合用，就能作出正电继电器ZDJ电路，如图10—31所示。

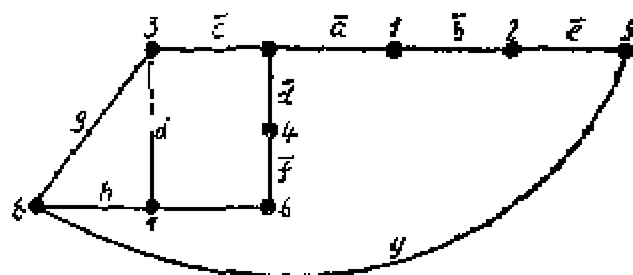


图 10—30

在一个单接点开关函数的六步合成法中，我们看到第一、二、四和六步是容易进行的，而且对F_{ij}不需要有什么先决条件。第三步比较麻烦，但是也能保证达到。用模2消除法可以把秩数k的任何一个矩阵变成下列形式之一。

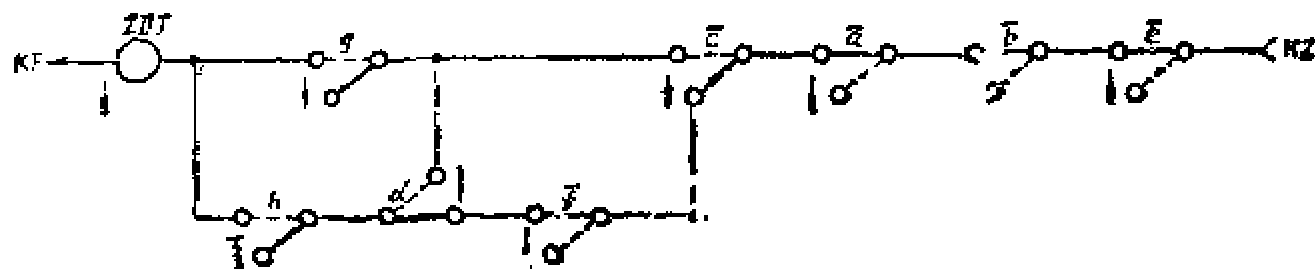


图 10—31

$$
\left(\begin{array}{c} U_k \\ \hline 0 \end{array}\right),\quad [U_k \mid 0],\quad \left(\begin{array}{c|c} U_k & 0 \\ \hline 0 & 0 \end{array}\right),\quad 或U_k \qquad (10—30)
$$

对于给出的开关函数是否是一个单接点函数，上述的第1，2，3和4步总是可以进行的。如果F不是一个单接点函数，到了第5步就会失败，因为用初等行运算，我们决不会得到一个缩减关联矩阵。这就是为什么我们要在二一节中先讲基本截割组矩阵的可实现性了。

第六节　不定开关函数的单接点网络合成

一个不定开关函数是指一个开关函数的真值表中有些值是没有确定的。例如，下列的真值表

x_2	x_1	F
0	0	0
0	1	λ
1	0	1
1	0	1

式中符号λ表示F值是没有确定的。

对于一个不定开关函数F_{ij}，符号$\{R_{ij}\}$和$\{\lambda_{ij}\}$表示对应F_{ij}的两个组，它们满足：

（一）
$$F_{ij}(1)=\sum_{R_r\in\{P_{ij}\}} R_r \text{ 乘积} \qquad (10—31)$$

（二）
$$F_{ij}(2)=\sum_{\lambda_r\in\{\lambda_{ij}\}} \lambda_r \text{ 乘积} \qquad (10—32)$$

式中$F_{ij}(1)$是把F_{ij}中对应的真值表中所有λ置于0，和$F_{ij}(\lambda)$是把F_{ij}中对应的真值表中所有1置于0，以及所有λ置于1。

注意当我们把对应真值表中所有λ置于1时，

$$F_{ij}=F_{ij}(1)+F_{ij}(\lambda) \qquad (10—33)$$

我们也可以看出

$$\{R_{ij}\}\cap\{x_{ij}\}=\phi \qquad (10—34)$$

我们可以用$F_{ij}(1)$和$F_{ij}(\lambda)$而不用真值表来说明一个不定开关函数的。例如，我们用

$$F_{ij}(1)=x_2$$

$$F_{ij}(\lambda)=x_1\overline{x_2}$$

给出上面真值表所表示的一个不定开关函数，在这组方程中的集体$\{R_{ij}\}$和$\{\lambda_{ij}\}$是

$$\{R_{ij}\}=(x_2)$$
$$\{\lambda_{ij}\}=(x_1,\overline{x_2})$$

当一个不定开关函数给出时，我们可以选择$\{P_{ij}\}$，使

$$\{R_{ij}\}\subset\{P_{ij}\}$$

以及任何一个$\lambda\in\{\lambda_{ij}\}$可以在$\{P_{ij}\}$之中。可是，$\{P_{ij}\}$的选择一定要能按照上一节所讲的方法合成出一个单接点网络。下一例于说明选择集体$\{P_{ij}\}$的方法。

例10—18　有一个不定开关函数，它的集体$\{R_{ij}\}$和$\{\lambda_{ij}\}$为

$$\{R_{ij}\}=\{(a,e,f),(b,d,f),(c,d,e),(a,b,c)\}$$
$$\{\lambda_{ij}\}=\{(d,e,f),(a,b,f),(a,c,e),(b,c,d),(a,b,e)\}$$

假使我们选择$\{P_{ij}\}=\{R_{ij}\}$，这种选择将满足定理10—24。选$P_1=(a,e,f)$，$P_2=(b,d,f)$，$P_3=(c,d,e)$，得

$$B=\begin{matrix} (y)\cup P_1 \\ (y)\cup P_2 \\ (y)\cup P_3 \end{matrix}\overset{\begin{matrix}a & b & c & & d & e & f & y\end{matrix}}{\begin{pmatrix} 1 & 0 & 0 & \vdots & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & \vdots & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & \vdots & 1 & 1 & 0 & 1 \end{pmatrix}}$$

因为这个矩阵已经是一个基本回路矩阵B_f，所以我们可以根据式（10—23）求C_f

$$C_f=\begin{array}{c} \\ 1\\2\\3\\4\end{array}\begin{array}{c}\begin{array}{ccccccc}a&b&c&d&e&f&y\end{array}\\ \left(\begin{array}{ccc|cccc}0&1&1&1&0&0&0\\1&0&1&0&1&0&0\\1&1&0&0&0&1&0\\1&1&1&0&0&0&1\end{array}\right)\end{array}$$

根据定理10—18，这个矩阵是$[N_0^t \mid U_0^t]$，不能实现为一个图的基本截割组矩阵。所以$F_{ij}(1)$不是一个单接点函数。

现在假使我们在$\{P_{ij}\}$中加入$\{d,e,f\}$，那么

$$(d,e,f)\oplus P_1\oplus P_2=(a,b,f)\in\{\lambda_{ij}\}$$

根据定理10—24，(a,b,f)一定是在$\{P_{ij}\}$之中，同样

$$(d,e,f)\ominus P_1\oplus P_3=(a,c,e)\in\{\lambda_{ij}\}$$

表明(a,c,e)必须选在$\{P_{ij}\}$之中，最后

$$(d,e,f)\oplus P_2\oplus P_3=(b,c,d)\in\{\lambda_{ij}\}$$

表明(b,c,d)一定是在$\{P_{ij}\}$之中。因此我们有

$$\{P_{ij}\}=\{R_{ij}\}\cup\{(d,e,f),(a,b,f),(a,c,e),(b,c,d)\}$$

很明显，这个集体满足定理10—24。选择$P_4=(d,e,f)$也作为一个线性独立路径，我们则有

$$B=\begin{array}{c} \\ (y)\cup P_1\\(y)\cup P_2\\(y)\cup P_3\\(y)\cup P_4\end{array}\begin{array}{c}\begin{array}{ccccccc}a&b&c&d&e&f&y\end{array}\\ \left(\begin{array}{ccccccc}1&0&0&0&1&1&1\\0&1&0&1&0&1&1\\0&0&1&1&1&0&1\\0&0&0&1&1&1&1\end{array}\right)\end{array}$$

将行4加到行2和行3，得

$$B_f=\begin{array}{c}\begin{array}{ccccccc}a&b&c&d&e&f&y\end{array}\\ \left(\begin{array}{cccc|ccc}1&0&0&0&1&1&1\\0&1&0&0&1&0&0\\0&0&1&0&0&1&0\\0&0&0&1&1&1&1\end{array}\right)\end{array}$$

由此得一个基本截割组矩阵C_f

$$C_f=\begin{array}{c}\begin{array}{ccccccc}a&b&c&d&e&f&y\end{array}\\ \left(\begin{array}{cccc|ccc}1&1&0&1&1&0&0\\1&0&1&1&0&1&0\\1&0&0&1&0&0&1\end{array}\right)\end{array}$$

将行3加到行2，得

$$C=\begin{array}{c}\begin{array}{ccccccc}a&b&c&d&e&f&y\end{array}\\ \left(\begin{array}{ccccccc}1&1&0&1&1&0&0\\0&0&1&0&0&1&0\\1&0&0&1&0&0&1\end{array}\right)\end{array}$$

最后一个矩阵每列最多有两个1，因此我们可以构成图G，如图10—32所示。删除边y，即

删除虚线，我们得到满足F_{ij}的单接点网络。

如果我们不是选择 (d,e,f) 而是选择 (a,b,e) 作为$\{P_{ij}\}$的一个元素，就是

$$\{P_{ij}\}\supset\{R_{ij}\}\cup(a,b,e)$$

那么，

$$(a,b,e)\oplus P_1\oplus P_2=(d)\notin\{\lambda_{ij}\}$$

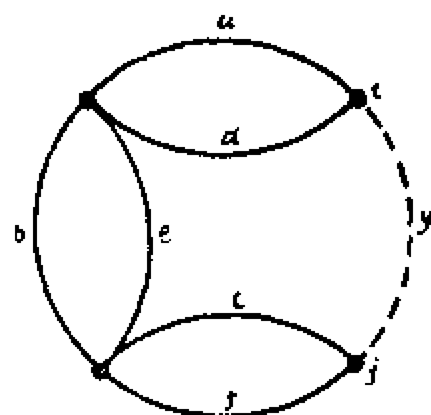


图 10—32

因此在$\{P_{ij}\}$中再加入$\{\lambda_{ij}\}$中任何其它元素是不可能满足定理10—24的。所以，选择(d,e,f)是得到一个所需的单接点网络的唯一途径。

这里一定要注意的是对$\{P_{ij}\}$的不同选择，只要在$\{P_{ij}\}$中所出现的不同布尔变数的个数是相同的，则所得到的单接点网络中的边数是不变的。换句话说，一个单接点网络中的边数是由$\{P_{ij}\}$中不同布尔变数的个数决定的。

第七节 多接点开关网络的合成

在单接点网络的合成中，每边的权一定是一个不同的布尔变数。但是在多接点开关网络中，一个布尔变数可以是两条边或更多边的权。

多接点开关网络有两种。一种是一个布尔变数可以是开关网络中两条边的权，可是，如果两条边有同一布尔变数x时，那么一个一定是x，而另一个一定是它的补数$\overline{x}$。这种开关网络在接点网络中叫作前后接点或相依接点开关网络。另一种多接点开关网络是一个布尔变数可以出现在网络中的几条边中，如图10—24所示。

前后接点开关网络是一种无向图，它的特点是（1）每一条边的权是一个布尔变数，（2）对于每一个布尔变数，最多有一条边的权是x，和最多有一条边的权是$\overline{x}$。

设有一个完全确定了的前后接点开关函数F_{ij}表达为

$$F_{ij}=\sum_{P_r\in\{P_{ij}\}}P_r\text{ 乘积} \tag{10—35}$$

集体$\{P_{ij}\}$不一定是图G中点i和j之间的所有可能路径的集体。这是因为图中可能有这样一条路径，它包括两条边，而这两条边的权是互补的，象x和$\overline{x}$。这样就因为$x\overline{x}=0$，所以在F_{ij}中不会对应于一个非零项。例如，图10—33所示的一个前后接点网络的F_{ij}为

$$F_{ij}=x\overline{y}+yz+\overline{x}\,\overline{y}z$$

对应的$\{P_{ij}\}$为

$$\{P_{ij}\}=\{(x,\overline{y}),(y,z),(\overline{x},\overline{y},z)\}$$

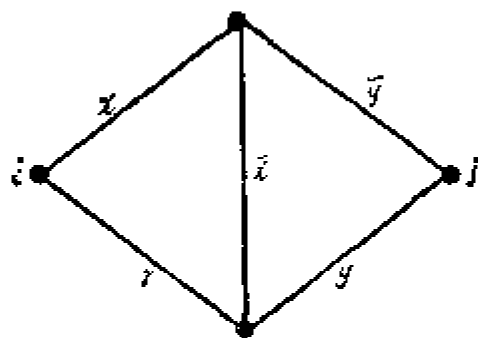


图 10—33

很明显，这不是图10—33所示网络中点i和j之间所有可能路径的集体。一条包括布尔变数和它的补数的路径叫作O-路径，而图10—33中就有这样的O-路径，例如路径$(x,\overline{x},y)$。

现在可以弄清楚，对应一个前后接点开关网络的函数F_{ij}一般不能满足定理10—24。因此，为了要按照本章第十节中所讲的步骤来合成一个前后接点开关网络，我们要在集体$\{P_{ij}\}$中加入O-路径（在点i和j之间的路径）使成为$\{P'_{ij}\}$后才能进行。

例10—19　设有一个开关函数F_{ij}如下：

$$F_{ij}=ace+ac\overline{d}\,\overline{e}+\overline{a}bce+bd\overline{e}+\overline{a}bc\overline{d}\,\overline{e}$$

对应F_{ij}的$\{P_{ij}\}$为

$$\{P_{ij}\}=\{(a,c,e),(a,c,\overline{d},\overline{e}),(\overline{a},b,c,e),(b,d,e),(\overline{a},b,c,\overline{d},\overline{e})\}$$

这个集体因为：

$$(a,c,e)\oplus(a,c,\overline{d},\overline{e})\oplus(b,d,\overline{e})=(b,d,\overline{d}e)\notin\{P_{ij}\}$$

$$(a,c,e)\oplus(\overline{a},b,c,e)\oplus(b,d,\overline{e})=(a,\overline{a},d,\overline{e})\notin\{P_{ij}\}$$

$$(a,c,\overline{d},\overline{e})\oplus(\overline{a},b,c,e)\oplus(b,d,\overline{e})=(a,\overline{a},d,\overline{d},e)\notin\{P_{ij}\}$$

所以不能满足定理10—24。可是，这些环和所得的结果路径都是些O-路径，因此我们可以把它们增加到$\{P_{ij}\}$中去而不致改变F_{ij}的性质。所得的$\{P'_{ij}\}$为

$$\{P'_{ij}\}=\{(a,c,e),(a,c,\overline{d},\overline{e}),(\overline{a},b,c,e),(b,d,\overline{e}),$$
$$(\overline{a},b,c,\overline{d},\overline{e}),(b,d,\overline{d},e),(a,\overline{a},d,\overline{e}),(a,\overline{a},d,\overline{d},e)\},$$

设$P_1=(a,c,e)$，$P_2=(a,c,\overline{d},\overline{e})$，$P_3=(\overline{a},b,c,e)$，$P_4=(b,d,\overline{e})$为所选的线性独立路径，我们就可以得到一个回路矩阵$B$为：

$$B=\begin{array}{c}\begin{array}{ccccccccc}a&\overline{d}&\overline{a}&d&b&c&e&\overline{e}&y\end{array}\\\left(\begin{array}{ccccccccc}1&0&0&0&0&1&1&0&1\\1&1&0&0&0&1&0&1&1\\0&0&1&0&1&1&1&0&1\\0&0&0&1&1&0&0&1&1\end{array}\right)\end{array}$$

将行1加到行2，得

$$B_f=\begin{array}{c}\begin{array}{cccc|ccccc}a&\overline{d}&\overline{a}&d&b&c&e&\overline{e}&y\end{array}\\\left(\begin{array}{cccc|ccccc}1&0&0&0&0&1&1&0&1\\0&1&0&0&0&0&1&1&0\\0&0&1&0&1&1&1&0&1\\0&0&0&1&1&0&0&1&1\end{array}\right)\end{array}$$

基本截割组矩阵C_f为

$$C_f=\begin{array}{c}\begin{array}{cccc|ccccc}a&\overline{d}&\overline{a}&d&b&c&e&\overline{e}&y\end{array}\\\left(\begin{array}{cccc|ccccc}0&0&1&1&1&0&0&0&0\\1&0&1&0&0&1&0&0&0\\1&1&1&0&0&0&1&0&0\\0&1&0&1&0&0&0&1&0\\1&0&1&1&0&0&0&0&1\end{array}\right)\end{array}$$

将行1加到行5，并将行2加到行3，得一个缩减关联矩阵A_f为

$$A_f=\begin{array}{c}\begin{array}{ccccccccc}a&\overline{d}&\overline{a}&d&b&c&e&\overline{e}&y\end{array}\\\left(\begin{array}{ccccccccc}0&0&1&1&1&0&0&0&0\\1&0&1&0&0&1&0&0&0\\0&1&0&0&0&1&1&0&0\\0&1&0&1&0&0&0&1&0\\1&0&0&0&1&0&0&0&1\end{array}\right)\end{array}$$

从这个缩减关联矩阵 A_f，我们构成图G如图 10—34 所示。删除那条虚线表示的边y，这个图就是所需要合成的前后接点开关网络。

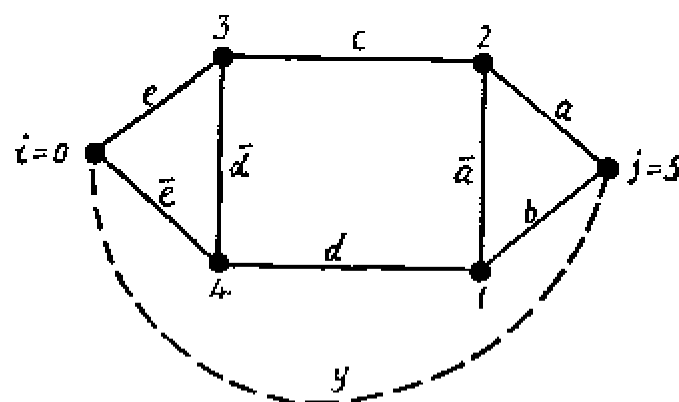


图 10—34

为了合成一个前后接点开关网络，如果所给出的是一个不定开关函数时，我们要按照上一节的办法把函数分成两个集体即$\{R_{ij}\}$和$\{\lambda_{ij}\}$，然后在$\{P_{ij}\}$中增加些O-路径。其余可以完全照上一节的步骤进行来求取图G。

另一种多接点开关网络的合成，如果照本章第五节的办法进行，那就变得很复杂了。为了仍旧应用这套办法，我们要给每一条边一个不同的名称，这样在一个给出的开关函数 F_{ij}中就很难判断一个布尔变数是否属于一条边。例如，在对应一个开关函数F_{ij}的集体$\{P_{ij}\}$中的 (a, b, c) 也许可能从 (a_1, a_2, b, c) 中来的，其中$a_1=a_2$。换句话说，假使在开关网络的点 i 和 j 之间有一条路径，它包括有同一权 x 的两条边，那么这条路径在$\{P_{ij}\}$中的P乘积中只有F_{ij}中的一个 x。当我们想应用第五节的办法时，我们一定要改变这条路径使它的两条边有不同的布尔变数。这就不是一件容易的工作。下面的例子说明这个事实。

例10—20 有一个开关函数F_{ij}为

$$F_{ij}=ab+ac+bc$$

对应这个函数F_{ij}的集体$\{P_{ij}\}$为

$$\{P_{ij}\}=\{(a, b)(a, c), (b, c)\}$$

因为

$$(a, b)\ominus(a, c)\oplus(b, c)=\phi$$

这个集体$\{P_{ij}\}$不能满足定理10—24。如果我们改变

$$(a, b)\longrightarrow(a_1. b_1)$$
$$(a, c)\longrightarrow(a_2, c)$$
$$(b, c)\longrightarrow(b_2, c)$$

式中 $a_1=a_2 \quad b_1=b_2$

则

$$(a_1, b_1)\oplus(a_2, c)\oplus(b_2, c)=(a_1, a_2, b_1, b_2)$$

所以把集体$\{P_{ij}\}$增加为$\{P'_{ij}\}$，即

$$\{P'_{ij}\}=\{(a_1, b), (a_2, c), (b_2, c), (a_1, a_2, b_1, b_2)\}$$

就能满足定理10—24。第五节中的方法就可以应用到这里来构图了。

第八节 时序开关网络

上面所讲的都是组合开关网络。现在我们要讨论时序开关网络，也有把它叫作序贯机的。上面已经讲过，时序开关网络的输出不仅根据现在的输入，而且也根据过去的历史。所以一个时序开关网络一定要保持着关于过去输入的信息。这就引入一个时序开关网络的“状态”概念。这里所谓状态就相当于过去输入的记忆。在数学方面，一个时序开关网络是如下定义的：

一个时序开关网络是一个数学系统M，它包括：

（一）一个内部状态（或简称状态）的有限组$V=\{v_1, v_2,\cdots, v_n\}$。

（二）一个输入有限组$X=\{x_1,x_2,\cdots,x_m\}$，叫作输入字母。

（三）一个输出有限组$Z=\{z_1,z_2,\cdots z_p\}$，叫作输出字母。

（四）一个函数或映射，它对现在状态和现在输入的每一个组合（v_i，x_j）指定它下一个状态v_k。这个函数叫作M的转换函数。

（五）另一个函数叫作输出函数，它对现在状态和现在输入的每一个组合（v_i,x_j）指定一个输出Z_i。

说明一个时序开关网络有两种等效的方法：（1）叫做状态表的表格方式，（2）叫做状态图的加权有向图方式。在状态图中，每一个点对应着时序开关网络的一个状态，每一条有向边代表从现在状态到次一状态的转换。对每一条边（v_i，v_j）指定给它一对有序权(x_k，z_q）或称有序权对。这一权对代表着这样一个事实，就是，如果网络的现在状态是v_i，并且现在输入是x_k，那么输出是z_q，并且次一状态是v_j。图10—35是一个时序开关网络的状态图，表10—1是它的状态表，这个网络有下列数据：

状态　$V=\{A,B,C,D\}$

输入　$X=\{1, 2\}$

输出　$Z=\{a,b,c\}$

在这表中可以看出，例如，当开关网络是在状态A时，当输入为1时，网络的次一状态为B，产生一个输出a，而当输入为2时，网络的次一状态仍为A，产生一个输出b。余类推。

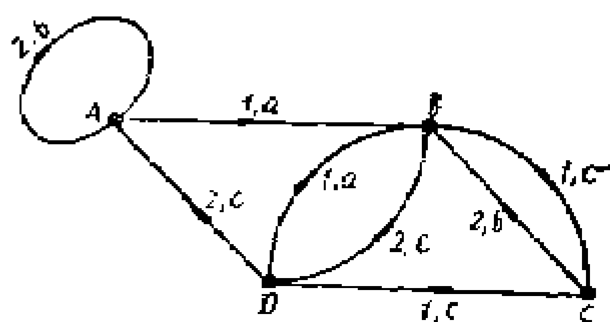


图　10—35

状　态　表　　表10—1

现在状态	次一状态/输出	
	当输入为1时	当输入为2时
A	B/a	A/b
B	C/c	D/c
C	D/c	B/b
D	B/a	A/c

状态图有下列特性：

（一）对每一个规定输入，在某一规定状态的网络响应着进入次一特定状态。所以，每一个点的出次数为m，即每一个输入的出次数为1；并且状态图有nm条边。注意对于入次数是没有类似的约束。

（二）因为一个输入可能使时序开关网络留在现在状态，所以状态图中可能有自环。

（三）一个状态图也可能有平行边；不过它们的权对是不同的。

（四）在大多数情况下，时序开关网络中有一个状态指定为起动状态，并且要求网络在施加任何输入之前是在这个状态的。有一个指定起动状态的网络的状态图可以看作是一个有根有向图，根就是起动状态。

（五）不管哪一个输入施加到这个网络，而它的状态（如果有的话）不会改变的，这个状态叫作持久状态。如果一个时序开关网络有一个持久状态时，从这个对应点就没有有向边通往另一个点了。

（六）如果一个时序开关网络是强连通的，这个时序开关网络就叫作强连通的。当且仅当一个时序开关网络M可以用一个适当的输入序列从任何一个状态引至任何另一个状态时，这个网络M是强连通的。

一个时序网络的状态图包含着关于这个网络的所有信息。因此研究一个给定网络的状态图包含着关于这个网络的所有信息。因此研究一个给定网络的状态图就可以研究这个网络的特性。时序开关网络理论中所出现的一些问题是：

（一）分析：在分析一个时序开关网络时，我们的兴趣可能在于一个给出的网络对于一定输入序列的响应（即次一状态和输出），或者我们的兴趣可能在于施加一系列输入时对网络内部作用的结论，并观察它的输出。如果一个网络有一个指定的起动状态，由于施加某一输入序列，结果是一个唯一的输出序列。

（二）合成：在设计一个给定要求作用的时序开关网络时，我们要根据所希望响应的陈述先构成一个状态图。注意下面的例子。

1．问题：对应一个任意的0和1的输入序列，设计一个时序开关网络。当每次有一串8—4—2—1二进制编码的十进制码（最低位首先进入网络）4个连续位的值大于9输入时，网络的输出总是1。当每次4位序列的值等于或小于9（即0000，0001、0010，…，1001）输入时，网络的输出总是0。

2．解法：网络应能存储最后的3个连续位，并应能检查和响应次一数位。因此，我们应该从一个8-状态（$2^3=8$）时序开关网络开始。设8个状态000，001，010，011，100，101，110和111分别命名为A，B，C，D，E，F，G和H。输入字母包括{0，1}，输出字母也包括{0，1}。

当一个新的数位到达状态图的左端，即输入端时，网络丢去最右位，并把新进入的位与两个旧位一同存储起来。例如，如果网络是在状态C（即010），并且进入一个1时，次一状态是101（即F），输出是1（对应着1010）。这样一个时序开关网络的状态表和状态图是容易构成的，图10—36所示是它的状态图，表10—2是它的状态表。

状态表 表10—2

现在状态	次一状态/输出	
	0	1
A	$A/0$	$E/0$
B	$A/0$	$E/0$
C	$B/0$	$F/1$
D	$B/0$	$F/1$
E	$C/0$	$G/1$
F	$C/0$	$G/1$
G	$D/0$	$H/1$
H	$D/0$	$H/1$

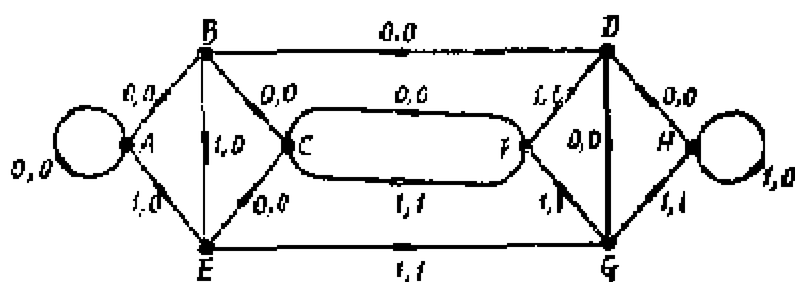


图 10—36

（三）状态等效和状态简化：我们刚才得到的8-状态网络不一定是执行指定任务最简单的一种。下一步也是很重要的一步是检查状态图，并且看看能否简化成一个比较简单些的网络。所谓简化就是在一个给定的网络的状态图中的一对点之间，确定两个状态是否等效。如果两个状态产生相同的输出，并且对每一个输入也进至一对相同的次一状态时，这两个状态就可以看作是一个状态；不论它们在状态表中出现在什么地方，都可以给予同一标号的。

在状态图中，我们可以合并两个等效点，并且在合并过程中删除掉任何冗余边（有相同权的平行边）。在图10—36中，点A和B是等效的，所以这两点可以合并。这种合并的结果

就成为 7 -状态网络，如图10—37(a)所示。

简化过程见图10—37。简化完成时所得的状态图如图10—37（e）所示。图（e）中的状态 A 取代了原状态图中的 A 和 B，状态 C 取代了原来的 C 和 D，状态 E 取代了原来的 E，F，G 和 H。图（e）所示的 3 -状态时序开关网络可以完成图10—36中原来的 8 -状态网络的同样任务。这个简单的例子说明了状态简化的重要性。

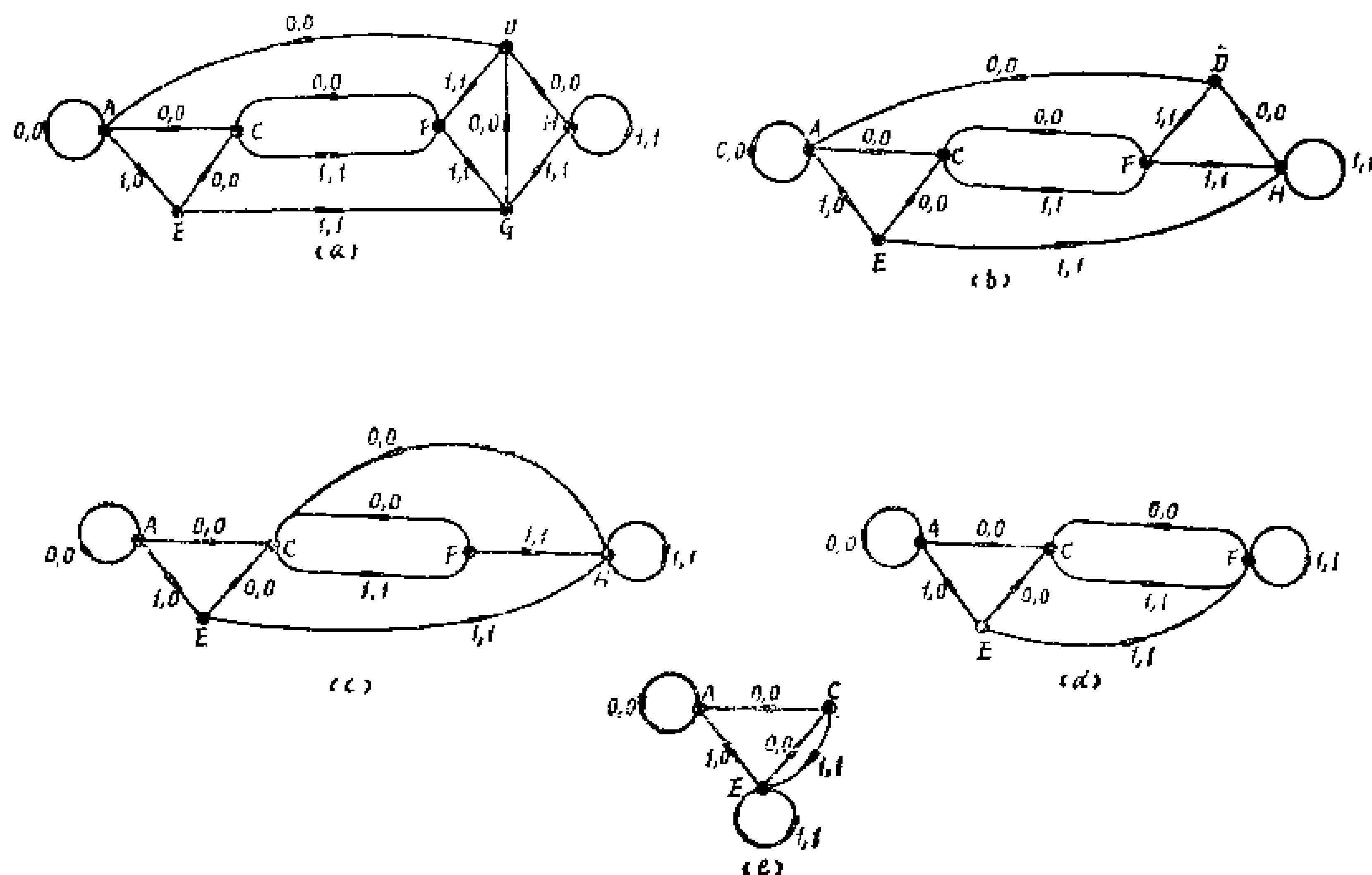


图 10—37

（四）状态分配：有了简化的状态图后，下一步是时序开关网络的实现。假设我们使用二进制记忆器件，如像触发器或拨扭开关等，一个 n -状态网络将需要 q 个这种器件，其中 $2^{q-1} < n \leqslant 2^q$，$q$ 个二进制记忆器件可以有 2^q 个可能状态。如何把这些 2^q 个状态分配到状态图的 n 个点去使我们得到最经济的网络，这就是状态分配的问题。

在图论术语中，状态分配就是用某种最优准则，把可利用的 2^q（$\geqslant n$）个标号指定给一个 n -点有向图中的各点。

在时序开关网络理论中，找出一个有效算法来得到最佳分配是一个重要的尚未解决的问题。如果先列出各种可能的分配方案，再选择其中最好的一个，既使对于只有10个状态的网络也是做不到的。可是对于一个很小的网络，如图10—37（e）所示的 3 -状态网络，这是可能弄明白各种不同分配方案，并加以比较的。当 $n = 3$，$q = 2$ 时，不同分配方案是 3 。下列的表10— 3 就表示这 3 种不同分配方案，其中 y_1 和 y_2 是两个记忆器件。

如果你熟悉逻辑器件，可以完全用触发器（或延迟线）和门电路来设计这些时序开关网络的。你会看到一个分配方案无疑比其它两个为优越。

在时序开关网络理论中，还有若干很重要但是很困难的问题等待解决，图论有可能来解决这些问题中的若干个。

分 配 方 案 表　　　　表10—3

状态	分配方案 1		分配方案 2		分配方案 3	
	y_1	y_2	y_1	y_2	y_1	y_2
A	0	0	0	0	0	0
C	0	1	0	1	1	1
E	1	0	1	1	1	0

习　　题

10—1　应用路径集体求图10—38中开关函数F_{12}和F_{13}。

10—2　应用接通矩阵求图10—39的开关函数F_{12}。

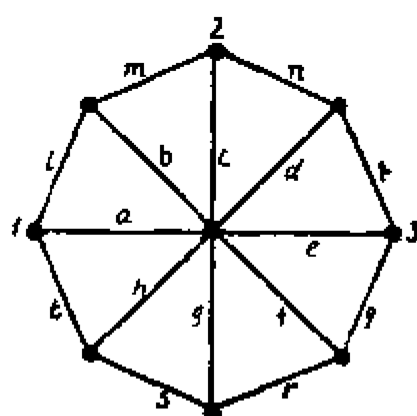


图　10—38

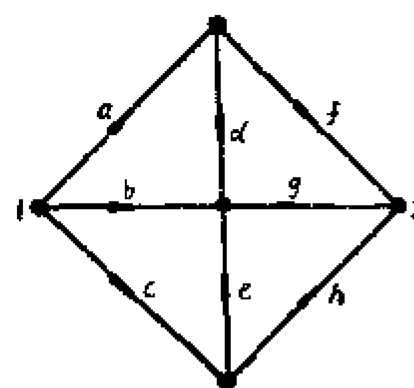


图　10—39

10—3　合成下列开关函数

$$F = af + bdf + cdef + cg + beg + adeg$$

10—4　有一个不定开关函数，它的集体$\{R_{ij}\}$和$\{\lambda_{ij}\}$为

$$\{R_{ij}\} = \{(a, d, e), (b, d, f), (b, e), (c)\}$$

$$\{\lambda_{ij}\} = \{(a, c, e\}, (a, f)\}$$

试合成一个单接点网络。

10—5　以下列开关函数合成一个多接点开关网络

$$F_{ij} = \overline{xy}z + \overline{x}yz + \overline{yuz} + \overline{y}u\overline{z}$$

10—6　下列函数是一个单接点开关函数吗?

$$F_{ij} = ab + ac + \bar{b}c + ad$$

10—7　如果F_1是一个单接点网络的完全确定了的开关函数，那么$F = xF_1 + \overline{x}F_1$是一个前后接点网络的完全确定了的开关函数吗?

第十一章　图论算法和计算机程序

第一节　引　　言

学习图论的一个重要部分，尤其是那些着眼于应用方面的人们，是使用数字计算机来解决图论问题。应用图论来解决的实际问题，如果是些小问题，就是比较简单线性图的问题，我们用手算也是可以解决的。这在前几章中已经提到几种算法。即使是简单的小的线性图，手算起来还是比较费时的，可是大部分的实际问题都牵涉到一些大的线性图，这些图实在是无法用手算的。事实上，近年来人们所以对图论大感兴趣的原因之一就是高速电子计算机的出现。所以迄今还只是单纯理论上空谈的问题现在突然可以用计算机来解决了。并且它们的解决可以应用到实际场所。计算机程序编写出来可以成功地处理电网络，电路设计、交通网络，流量问题、计划评审技术、以及其它问题中所出现的大的线性图。

可是，我们必须看到，虽然现在的计算机的运算速度很快，能工作于毫微秒（10^{-9}秒），如果作为一个有巨大力量的工具来解决图论问题，事实上，任何组合学问题，很快它们就会达到限度而不够用了。例如，在一个有n_v个点的加权完全图中，为了寻找最低权汉密尔顿回路，就是第二章第十三节中巡回检查工作的旅程问题，就有$\left[\frac{1}{2}(n_v-1)!\right]$个不同的汉密尔顿回路，人们也许想试用有力工具来产生所有汉密尔顿回路，并加以比较各个回路的权。设想有一个只有10个点的图，所有点的汉密尔顿回路就有$\frac{1}{2}(10-1)!=181440$，用计算机来计算也许会行。但是设想一个有20个点的图，就有

$$\frac{1}{2}(20-1)!\approx 6\times 10^{16}\text{个回路}$$

即使每次运算只要用1毫微秒，要运算6×10^{16}次就会需用约

$$\frac{6\times 10^{16}}{10^{9}\times 3\times 10^{7}}=2\text{年}$$

这就可以清楚地看出，不管电子计算机的速度怎样快，如果没有数学工具的帮助，人们是不能想像能够得到这个所需要的数值答案的。所以一定要把计算机的力量与数学方法的技巧相结合起来的。

像在所有的组合学问题一样，图和子图的处理和分析，本质上不是数值的。在图论程序中，基本上是使用计算机的判定能力而不是它的算术运算能力。

学习这一章内容之前，应该先有计算机程序设计基础。

第二节　算　　法

一种算法实质上是解决某一数学问题的一个处方，它包括一套指令，一步一步地按照它进行，就能引导到问题的解决。在一个算法中的每一步必须规定得精确和明白，并且一个算法在一定步数解决问题后必须结束。每一种算法必须要有5个特点：（1）有限性，（2）

明确性，（3）输入，（4）输出，（5）有效性。

一种算法可以用不同的方式表达出来的：（1）各步可以用英语写成；（2）可用所使用的计算机能懂的语言详尽地写成计算机程序方式；或（3）前两者的中间方式，即用框图方式。每一种方式各有优缺点。通常一种算法先用普通语言表达出来，然后变成框图，最后写成详细和精确语言，以便计算机执行。

便于说明问题，框图是最好的方式，这是表达一种算法的最通用的方式，它与程序设计语言和所使用的计算机无关的。在这一章的第五节中将附有几个实际程序的例子，列举出几个试验过的程序。

一种算法不仅一定要完成它所要做的任务，而且一定要有效地完成它。一种算法的效率有两个主要标准，就是作为输入大小的一个函数的存储量要求和计算时间要求。对我们来讲，输入就是一个线性图，而它的大小就是点数n_v和边数n_e。对大多数线性图问题来讲，存储量的要求一般不是关键，而计算时间正象上一节中所提到的，才是关键呢。

估计一种算法的优点，人们可能要寻找最坏条件的执行时间（就是对于某一给定大小的图的最坏可能选择所需要的执行时间），或最好条件的执行时间，或平均条件的执行时间。

对于解一个图论问题，常常不止一种算法可用。有时算法对于所有不太普通的线性图很容易看出是比其它算法为更有效些。可是在很多情况下，相对效率只能从图的大小和结构，算法的详细执行，和所使用的计算机这几方面来比较。

一种图论算法性能的详细分析是极为复杂的，我们不想在这方面多花时间。可是，我们对各种算法的复杂性一定要有一个总的意见，象假定输入的是最坏条件的线性图。当n_v或n_e变得很大时，计算时间如何作为n_v或n_e的函数而增加。这样一种性能指标对于估计一种程序的计算时间也许不太切合实际，没有什么用处的，不过在算法分类和它们的理论研究中常常是有价值的。

第三节　图在计算机中的表示法

一种算法有几个输入，它们是算法开始工作的依据，正象一盘菜的配料一样。很自然，我们的算法的输入是一个或几个无向图或有向图。一般讲，一个图在一个数字计算机中用下列5种方式之一来表示并存储。方式的选择是按照图、问题、语言、机器类型，以及在计算中图是否要变更而定。

（一）邻接矩阵

一个无向图或有向图输入到一个计算机的最通用方式是它的邻接矩阵$X(G)$。对一个图G的n_v个点中的每一个点各指定一个数，在输入、存储、和输出中就用$n_v \times n_v$二进制矩阵$X(G)$来表示G。因为n_v^2个元素值中的每一个或是0或是1，邻接矩阵需要n_v^2位的计算机存储器，数位可以组合成字。设w是字长（就是一个计算机字的位数），n_v是图的点数，那么邻接矩阵的每行可以是一系列的n_v位写成$\lceil n_v/w \rceil$个计算机字。符号$\lceil x \rceil$表示不少于x的最小整数。所以存储邻接矩阵所需要的字数为

$$n_v \lceil n_v/w \rceil \qquad (11-1)$$

一个无向图的邻接矩阵是对称的，因此只要存储上三角就够了，这只需要$n_v(n_v-1)/2$个存储位。存储量的节省却常常会带来了复杂性和计算时间的增加，可是有些问题是值得这样做的。

当图是稀疏时，就是图的边数n_e少于点数n_v的n_v^2时，邻接矩阵不是一种有效的表示法。必须注意，邻接矩阵的定义中规定没有平行边的，所以邻接矩阵是不能表示平行边的。

（二）关联矩阵

有时关联矩阵$A(G)$也用来存储并处理一个图。一个关联矩阵需要$n_v n_e$位存储量。因为一个图的边数n_e通常比点数n_v为多，所以这个方法所用的存储量可能比邻接矩阵所需的n_v^2位更多些。虽然存储量要大些，偶尔用关联矩阵要比用邻接矩阵有利些，而且关联矩阵容许有平行边的。关联矩阵特别有利于电网络和开关网络。

（三）列举各边法

经常使用的另一种表示法是把n_e条边任意编排，而列举图中每一对点之间的所有各边。例如，图11－1中的有向图可以把各边列举为下列点对：（1，2），（2，1），（2，4），（3，2），（3，3），（3，4），（4，1），（4，1），（5，2）。如这个图是无向，我们只要略去每一点对中的顺序。很清楚，这个图的表示法可以包括平行边和自环。

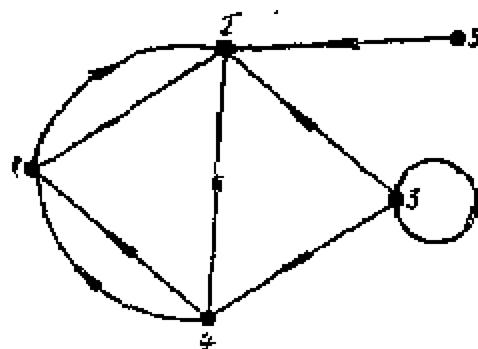


图 11－1

需要用以标记各点（由1到n_v）的位数为b，其中

$$2^{b-1} < n_v \leq 2^b \qquad (11-2)$$

因为n_e条边中的每一条边需要存储两个这种数，所以总存储量为

$$2n_e b\text{位} \qquad (11-3)$$

把这个量与n_v^2比较，如果

$$2n_e b < n_v^2$$

那么用列举各边法就比用邻接矩阵法为更经济些。换句话说，如果一个图的邻接矩阵是一个稀疏矩阵（所谓稀疏矩阵就是一个矩阵有很多元素值为零的，对邻接矩阵而言，就是n_e/n_v的比值为小的。）列举各边法是一种存储图的比较有效的方法。

列举各边法，是把一个图输入到计算机的一种很方便方式，但是在计算机中图的存储、检索和处理是相当困难的。例如，这就需要用广泛的检索技术来找寻一个图是否是连通的。这可以参看下面第五节中的算法一。

（四）两个线性数组法

与上述的列举各边法稍加变化的是用一个图的两个线性数组来表示。譬如说用$F=(f_1, f_2, \cdots, f_e)$和$H=(h_1, h_2, \cdots, h_e)$来表示一个图。在这些数组中的每一元素值是一个点的标记。如果G是一个有向图，第i条边e_i就是从点f_i到点h_i。如果G是无向图，e_i就是点f_i与点h_i之间的一条边。

例11－1　图11－1中有向图G的两个数组是：

$$F=(5, 2, 1, 3, 2, 4, 4, 3, 3)$$
$$H=(2, 1, 2, 2, 4, 1, 1, 4, 3)$$

这种表示法便于在加权图中分类。存储量与列举各边法相同。

（五）后继点列举法

如果一个图的n_e/n_v的比值不大的话，另一个常常用来输入图到计算机的有效方法是用n_v个线性是用n_v个线性数组。用任何顺序对图中各点指定一个数，如1，2，…，n_v后，每一个点k用一个线性数组来代表，每一个数组的第一个元素是k，而其余的元素是紧接k的那些后继点。就是在有向路径上离k只有一段距离的那些点，在无向图中，就只是邻接着k的那

些点了。

例11－2　图11—1中有5个点的有向图的后继点列举表示法。

1：2

2：1，4

3：2，3，4

4：1，1

5：2

对于一个无向图，就列举了每一个点的邻点而不是后继点。因此，每一条边出现两次，这是明显的冗余。

为了与邻接矩阵比较存储量，设d_{av}是图中各点的平均次数，在一个有向图中，则为平均出次数，假定每一个点的标记需用一个计算机字，对于一个有n_v个点的图的总存储量就要有$n_v(1+d_{av})$个字。所以，如果

$$d_{av}<\lceil n_v/w\rceil-1 \qquad (11—4)$$

式中w是字长。后继点列举法就比邻接矩阵更为有效。

后继点列举法，对于无向图就是邻点列举法，对于路径找寻算法和深度第一搜索法是极为方便的。

（六）关联清单

这方法与关联矩阵相似，不过这里不用图的关联矩阵而是把每一个点所关联的各边列成一个清单。这个清单只是一个简单的数组，就称为关联清单。我们还需要一个边表，它告诉我们各条边的两个端点，对于有向图来讲就是每条有向边的始点和终点。

找寻一条路径时，我们从关联清单中的第一条边的一个端点开端，先从边表中查出它的另一个端点，再从关联清单中找出这个新端点。就这样进行下去。

这个方法节省了从关联矩阵的行中搜索一个1的时间，当然这种节省只在n_v数较大和图为稀疏图时才是有效的。

人们一定已经观察到，上述线性图的各种表示法不是完全不同的。事实上有一点是互相有关的，就是它们都需要输送同样的信息。可以写出一个简单程序把一种方式变换成另一种方式，在这些表示法中的外加变化在不久的将来可以使它们适应要求。例如，一个加权图可以由$n_v\times n_v$权矩阵，也叫费用矩阵，或叫距离矩阵来代表。它与邻接矩阵相似，不过不用元素值而用各边所注的权。应该注意的是，在很多问题中，算法的效率可能根据表示那图的方式而不同。所以数据结构的适当选择是重要的。

第四节　图的输出

每一种算法有一个输出。输入是一个或几个图，而输出却视问题而不同。如果输出是一些子图，我们可以使程序打印出相当的邻接矩阵。另一方面，如果一个输出是问一个给定的图是否是平面的，我们可能要求程序只简单地打印出“是”或“否”。另外，如果答案是“是”，我们可能选择给出图的平面表示法；如果答案是“否”，我们可能要问图的厚度。对于一个最短路径算法，我们也许只希望打印出一对指定点x和y之间最短路径的边（或点）序列；……等等。所以算法不同，输出也不同。

下一节让我们着手些特定算法。

第五节　几种基本算法

算法一：连通度和部分算法

当我们碰到一个新图G时，第一个问题可能是问：G是连通的吗？如果G不是连通的，G的部分是什么？所以我们这里的第一种算法是决定一个给定图的连通度和部分的算法。

关于连通度和部分的问题，除了是算法本身的重要问题外，也是许多其它算法中提出的问题。例如，在测试一个图G与另一个图的可分性、平面性或同构性之前，为了效率起见，最好先确定G有哪些部分，然后把每个部分进行所要求的检查。连通度的算法是很基本的，可以作为更复杂的图论算法中的一个子程序。这里要提醒读者一下，虽然在画一个图时，人们可以看出一个图是否连通的，可是如果这个图是用其它方式表达出来时，那么连通度这一问题对计算机或人们并不是显而易见的。

给出一个图的邻接矩阵X后，试行置换X的行和相应的列并且检查它是否象式（5－72）所示的分块形式，就可能确定这个图是否是连通的。可是，这是一个效率差的方法，因为这里可能牵涉到$n_v!$次置换。一个更有效的方法是应用定理5－7的推论B，在矩阵

$$Y=X+X^2+\cdots+X^{n_v-1}$$

中检查有无零的元素值。这个方法也不是很有效的。因为要进行大量的矩阵乘法。下面是一个有效的算法。

这个算法基本步骤是邻接点的合并。我们从图中的某一个点开始，合并所有的邻接点，再由这个合并了的点开始，继续合并所有邻接点。这样合并的手续继续下去，一直到再没有其它点可以合并为止。这就表示一个连通图已经合并成为一个单独的点了。如果图中所有的点都合并掉了，这个图是连通的。否则，我们从另外一个部分的新点开始，进行合并操作。

在邻接矩阵，把点j合并到点i是应用逻辑“或”操作，就是用逻辑加法把行j加到行i以及把列j加到列i。（注意，在逻辑加法中，$1+0=0+1=1+1=1$，和$0+0=0$）。在矩阵中行j和列j就弃除了。如果有困难或者很费时间去弃除那些行和列，那么可以让那些行和列留在矩阵中，不过要留意在下次合并时不要再考虑它们了。

注意，通过合并，一个自环在主对角线上出现为1，但是平行边由于逻辑加法而自动成为一条边。当然，这些对于图的连通度是没有影响的。

在这个算法中，可能进行的最大合并数为n_v-1，其中n_v是点数。因为在每次合并中，最多进行n_v次逻辑加法，执行时间的上限是与$n_v(n_v-1)$成正比的。

在每一个部分中，适当地选择开始合并邻接点的那个点是会提高效率的。当然选择这样一个点是费时的。

用计算机解决一个问题时，在上机前，我们先要分析问题，研究它的解法，作出解法的框图。这里的框图与第九章第一节中所讲的流图在本质上相同，（在计算技术中，框图也有人译为流图或流程图）同是一种有向图。不过在表示形式上有所区别。这里的框图是解题逻辑的图形表示法。框图中的点用一种特殊符号（见表11－1）来代替，符号之间的有向边表示过程的相互关系。

连通度和部分算法的框图如图11－2所示。

曾有人用$FORTRAN$语言写出一个计算有50个点的图的连通度和部分的程序，在IBM-7044计算机上运算，执行时间约为9/60秒。

算法二：生成树算法

框 图 符 号 表11-1

符 号	名 称	意 义
（矩形）	过 程	一种过渡，一种操作，或是一组操作，存储分配，算术运算等。
（菱形）	判 定	根据某种判定或逻辑选择而改变的程序控制。
（输出框）	输 出	输出：如打印，显示，录制等。
（圆）	停 机	操作完毕停机。

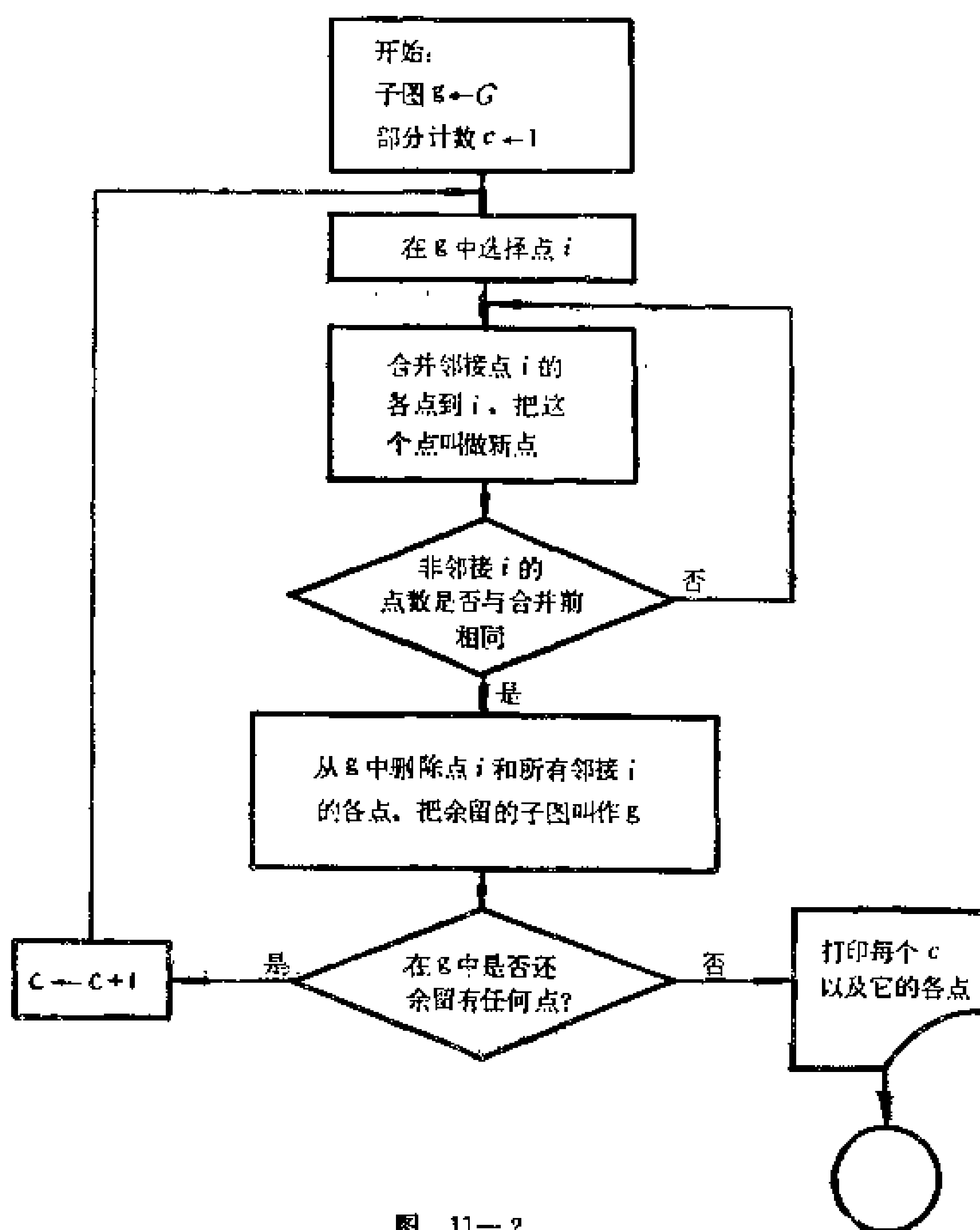


图 11—2

图论中最出名的和最常用的算法也许是生成树的算法，生成树算法的最简单形式是在一个给定的连通图中产生一棵生成树。如果图是分离图，算法会产生包括 $n-k$ 条边的一个生成林，其中 $k>1$ 是分离图中的部分数。所以很清楚，作为一个算法的副产品，我们可以检查出这个图是否是连通的；如果不是连通的，就鉴别它的部分。事实上，一个生成树算法有时可以用来测试一个图的连通度的。另一方面，如果给出的图的每边有权（即加权图），我们可能希望找出有最小可能权的一棵生成树。为了得到一个图的基本回路组，先要得到一棵生成树。为了产生一个图的所有生成树，也先要得到一棵生成树。

设一个给出的无自环（如果这个图有任何自环，可以抛弃它们）的无向图 G 包括 n_v 个点和 n_e 条边。设各点标以 $1,2,,\cdots,n_v$，而图用两个线性数组 F 和 H 来描述，即本章第三节的第四种图的表示法；两个线性数组输入法，f_i 和 h_i 是 G 中边 i 的端点，$f_i\in F$，$h_i\in H$。

在算法的每一阶段，要测试一条新边，看它的端点是否出现在到此为止已经组成的任何一棵树中。当然在起始阶段，图中是没有树的，最先考虑的第一条边（f，h）总是在一棵生成树（或林）之中的。所以用这个算法所产生的生成树（林）是与各边的编号法很有关系的。在第 p 阶段，$1\leqslant p\leqslant n_e$，测试边（$f_p$，$h_p$）时可能有下列五种情况：

（一）如果点 f_p 和 h_p 都不包括在 G 中到此为止已经组成的任何树中，这条边 p 称为一棵新树，它的端点 f_p，h_p 给以部分数 c，这个部分数 c 已经增加了一。

（二）如果点 f_p 是在某一棵树 T_i（$i=1,2\cdots c$）之中，并且点 h_p 是在另一棵树 T_j（$j=1,2,\cdots,c$，而 $i\neq j$）之中时，边 p 就用以连接这两棵树；因此，在 T_j 中的每个点现在给与 T_i 的部分数，这个部分数 c 现在减少了一。

（三）如果这两个点都在同一棵树中，边（f_p，h_p）组成一个基本回路，就不再考虑下去了。

（四）如果点 f_p 是在一棵树 T_i 中而 h_p 不在树中，边（f_p，h_p）加入 T_i，对 h_p 也指定以 T_i 的部分数。

（五）如果点 f_p 不在树中而 h_p 是在另一棵树 T_j 中，边（f_p，h_p）加入 T_j，对 f_p 也指定以 T_j 的部分数。

这五种情况在图11—3算法的框图中用加圈数字标出。

在这种算法的基础上，一种计算机程序的效率主要看一条边的端点是否出现在到此为止已经组成的任何树中的测试速度而定。为了这种测试，我们用一个大小为 n_v 的线性数组（在程序列表中称为 $VERTEX$）。当一条边（i，j）已经包括在树 c 中，这个数组的元素值 i 和 j 置位于 c。当另一条边（f_p，h_p）接着检查时，只要核对数组 $VERTEX$ 中 f_p 和 h_p 是否不是零。在数组中位置 q 的一个零表示点 q 到此为止还没有包括在任何一棵树中。执行结束时，这个数组 $VERTEX$ 鉴别图中各个部分。

一棵树不象一个部分那样，不能单独由点组来描述。因此，我们一定要有一个边的数组作为输出。设这个边的线性数组叫作 $EDGE$。如果边 p（按照这些边所安放的原来次序）是在树 c 之中，那么 $EDGE(p)=c$；否则这是零。在数组 $EDGE$ 中所有零元素对应着各弦（就是那些没有包括在生成树或林中的那些边）。用这个数组连同数组 F 和 H 来识别由这个算法所产生的生成树（林）。

在这个算法中，回线或称环路执行 n_e 次（n_e 是边数）。测试端点是否在任何一棵树出现的时间是一个常数，与 n_e 或 n_v 无关。因此用以执行这个算法的时间是与 n_e 成正比的。在比数 n_e/n_v 大的时候，执行时间可以引入一个新变数而减少，这个新变数用以计算树中所包括的

边数。当这变数达得到值n_0-1时，程序应终止（仅当图是连通的，否则，我们必须检查每一条边）。

这里附有一个用FORTRAN语言写的准备执行的程序。这是用FORTRAN Ⅳ语言写的子程序，在这个子程序中，输入用的点数不应大于100。

一个随便产生的有50个点的图，用FORTRAN Ⅳ语言编写的程序，在IBM-7044机上的执行时间只要2/60秒。

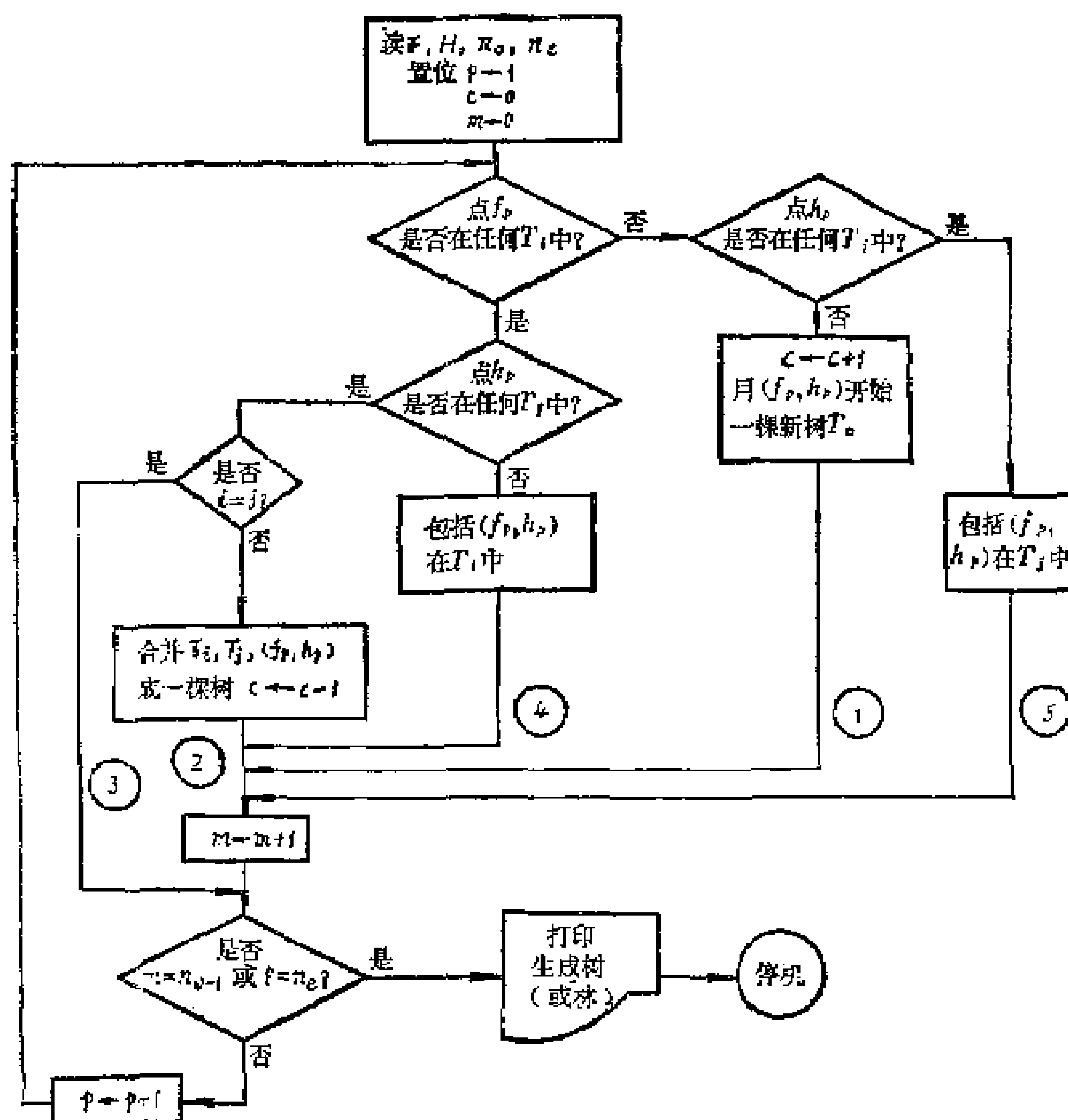


图 11—3

生成树（林）程序

```
      SUBROUTINE SPTREE (F,H,N,E,EDGE,C)
      INTEGER C, E, EDGE(E), F(E), H(E), VERTEX(100), V1, V2
      DO 4 L=1, N
4     VERTEX(L)=0
      DO 6 L=1, E
6     EDGE(L)=0
      C=0
      M=0
      K=0
10    K=K+1
```

```
      V1=F(K)
      I=VERTEX(V1)
      IF(I.EQ.O) GO TO 39
      V2=H(K)
      J=VERTEX(V2)
      IF (J.EQ.O) GO TO 36
      IF (I-J) 21, 50, 18
18    IJI=J
      J=I
      I=IJI
21    DO 26 L=1, N
      IF (VERTEX(L)-J) 26, 23, 25
23    VERTEX(L)=I
      GO TO 26
25    VERTEX(L)=VERTEX(L)-1
26    CONTINUE
      DO 32 L=1, E
      IF (EDGE(L)-J) 32, 29, 31
29    EDGE(L)=I
      GO TO 32
31    EDGE(L)=EDGE(L)-1
32    CONTINUE
      C=C-1
      EDGE(K)=I
      GO TO 49
36    EDGE(K)=I
      VERTEX(V2)=I
      GO TO 49
39    V2=H(K)
      J=VERTEX(V2)
      IF  (J.EQ.O) GO TO 45
      EDGE(K)=J
      VERTEX(V1)=J
      GO TO 49
45    C=C+1
      EDGE(K)=C
      VERTEX(V1)=C
      VERTEX(V2)=C
49    M=M+1
50    IF  (M.EQ. (N-1) .OR. K.EQ.E) RETURN
      GO TO 10
      END
```

在生成树算法中，有时不象上面所讲的只想找一棵生成树，而是要找一棵最短生成树，

或是要找一个图中的所有生成树。现在把这两种算法介绍于下。

（一）最短生成树算法

在第三章第九节中，我们已经讲过两个寻找最短生成树的算法。象图3－18所用的列表方法可用刚才讲过的生成树算法来完成的。所需增加的工作是在数组F和H代表各边之前先把它们照所注的权按递增次序予以分类。就是说，下列不等式必须满足：

边（f_i，h_i）的权≤边（f_{i+1}，h_{i+1}）的权，$1 \leqslant i \leqslant n_e - 1$。

由于牵涉到分类，这个算法不及该节中所讲的第二个算法的效率。第二个算法如图3－19所示，不必把各边分类，只要把部分组成的树连续接到最近的邻点来建成一棵最短生成树。这个算法的计算时间是与n_v^2成正比的，其中n_v是图的边数。

最短生成树对于第二章第十三节中巡回检查工作中的旅程问题中提供里程的下限是非常有用的。

（二）所有生成树算法

我们知道在电网络的分析中，从图论的观点看，主要是找寻图中所有的生成树。由于这样一个重要用处，已经有人提出十多种不同的算法来产生所有的生成树。在第三章第八节中，我们已经讲过一种方法，就是循环变换的方法。因为一个图，即使是一个小图的生成树数也是很多的。所以这种算法的效率是头等重要的事。有很多人曾对这种算法进行过研究，并且提出不同的算法。这里介绍一种最有效的算法于下。

这种算法的基本原理是：如果一个图G的一条边e_i是特异的，就是与其它边有所区别的，那么这图中的生成树可以分为两类。一类是包括这条边e_i的树，一类是不包括这条边e_i的树。如果G_1和G_2分别为从G中短捷边e_i为一个点和删除这条边e_i后所得的图，那么第一类生成树就是G_1加边e_i，而第二类生成树，如果边e_i既非一个自环，又非一条桥接边（就是这条边不是回路中的一个元素，或者这条边的移去能使图分离的），那么就是G_2了。

这个算法的步骤是：（1）先将给定的图画在一张纸上，所有各边都编好号或标以字母。（2）删除所有自环；删除所有的桥接边，把删除的边号记录在纸的下方。（3）利用这张纸按如下方法产生另外两张纸：取任何一条边叫它为e，短捷边e得到G_1，画在一张新纸上，并把边e的记号记录在纸的下方；删除边e得到G_2，画在另一张新纸上。（4）把原来那张纸下方记录的边号转载在这两张新纸的下方，原来的那张纸即可放弃。

现在对两张新纸的每一张照上法继续进行。每张纸又得到两张新纸。照这样继续下去，一直到原图上的所有边都处理完为止。这时，这张最新纸的下方将记载着生成树的各枝编号。把这些汇总，就得到原图的所有可能的生成树。

人们对于一个图的生成树有时有特殊的要求，譬如说，也许有兴趣于特殊约束的树，也许希望找寻一棵不是最短而是最长的生成树。前者将于第十二章第四节中讲到。

算法三：基本回路算法

有时候，我们需要找出一个给定图的一组基本回路。上面讲过的生成树算法，只要增加下列工作就可以用来产生基本回路的。

在生成树算法中，当检查边$p(f_p, h_p)$，如果出现第三种情况（就是，点f_p和h_p出现在同一棵树T_s）时，不要简单地抛弃这条边，而是一定要在T_s中点f_p和h_p之间找寻组成这条路径的那些边，这条路径和边（f_p，h_p）就组成一个基本回路。这里的主要问题是找寻这条路径。文献中有若干篇这方面的算法，现在把其中最有效的算法介绍于下。

它不象生成树算法中那样取任意次序的那些边，而是选择一个点 z（下面即将看到，点 z 是最近加到那棵部分构成树的那一个点），并检查所有关联到点 z 的每条边。设给定的连通图 $G=(V, E)$ 的各点标以 $1,2,\cdots,n_0$，并且这个图用邻接矩阵 X 给出。设 T 是部分构成树中新近加入的点组，并设 W 是还待检查的点组（就是那些在 T 中以及不在 T 中的那些点，关联到它们的一条边或几条边还未检查过的）。起初，$T=\phi$，$W=V$，就是整个点组。

算法的开始是设 $T=1$，即第一个点，和 $W=V$。点 1 将作为所组成的树的根。开始之后，就用下列步骤：

（一）如果 $T\cap W=\phi$，那么算法就终止。

（二）如果 $T\cap W\neq\phi$，在 $T\cap W$ 中选择一个点 z。

（三）检查点 z，考虑到所有关联到 z 的每条边。如果没有余留下这样的边，那么从 W 中除去 z，并转向第一步。

（四）如果 $Y\in T$ 时，找寻由边（z，r）时，测试点 r 是否在 T 中。

（五）如果 $Y\in T$ 时，找寻由边（z，r）以及树（到此为止已组成的树）中由 z 到 r 之间唯一路径所构成的基本回路。从图中删除边（z，r），并转向第三步。

（六）如果 $Y\notin T$ 时，把边（z，r）加入到树中去，并把点 r 加入到组 T 中去。从图中删除边（z，r），并转向第三步。

以前已经说到过，这个算法中唯一难办的部分是第五步，我们怎样去找寻树中由 z 到 r 的唯一路径呢？下列步骤可以回答这个问题。

我们准备一个后进先出表 $TW=T\cap W$。这是一个存储栈，它存储着树中还未检查过那些点。最新加入的点是在存储栈的顶部。每次要拿出一点来检查时就从这个存储栈的顶部去拿，并从存储栈中除去。这里用着两个段数为 n_0 的线性数组：$LEVEL$（1）是由生成树根（即由点 1）到点 i 的距离；$PRED(i)$ 是一个点 v，它在树的边（i，v）中，点 v 距根较近。换句话说，$PRED(i)$ 是由树根到 i 的路径中点 i 的前趋点（即前一个点）。当且仅当点 i 不在树的最新点组 T 中时，$LEVEL(i)=-1$。开始时，$LEVEL(1)$ 置于0，$LEVEL(i)$ 置于 -1，其中 $i=2,3,\cdots,n_0$。

在第五步中，点 z 正在检查中，找到一条边（z，r），使其中点 $z\in T$。为了找寻由边（z，r）和树所组成的基本回路，我们从树中点 z 到 r 连续找寻前趋点 $PRED(Z)$，$PRED(PRED(2))$，…，来追迹这唯一的路径，一至找到 r 的前趋点 $PRED(r)$ 为止。换句话说，基本回路是如图11—4所示那样产生的。

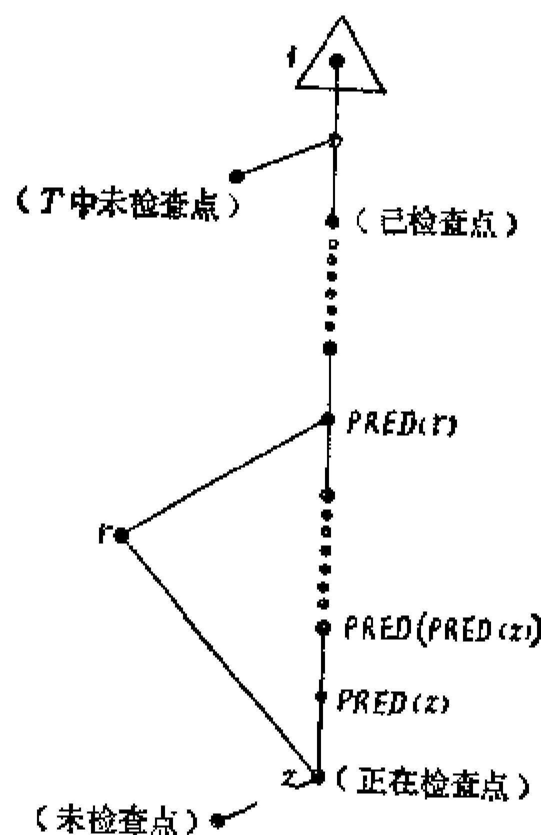


图 11—4

所产生的回路为：z，$PRED(z)$，$PRED(PRED(z))$，…，$PRED(r)$，r，z。

要注意的最重要的事情是在 T 中每一个点 p 的前趋点 $PRED(p)$ 是一个或者已经检查过的点，或者是正在检查的点。就是说 $P\in T\cap W$ 那么

$$PRED(p)\notin W，但是 PRED(p)\in T$$

图11—5是基本回路算法的框图：

这个算法的一个程序的执行时间受限于n_0^v的，$2\leqslant v\leqslant 3$，v值是由图的结构和各点的标记所决定的。

作为一个典型，一个有50个点和132条边的图，在IBM-7044机上用*FORTRAN* Ⅳ语言编写的程序操作，只费16/60秒就产生一组基本回路。

一个无向图中的所有回路：要找一个无向图中的所有回路可以先找出它的基本回路，然后构成一组全线性组合（就是环和），再从这组中删除所有包括其它回路的基本回路组。这

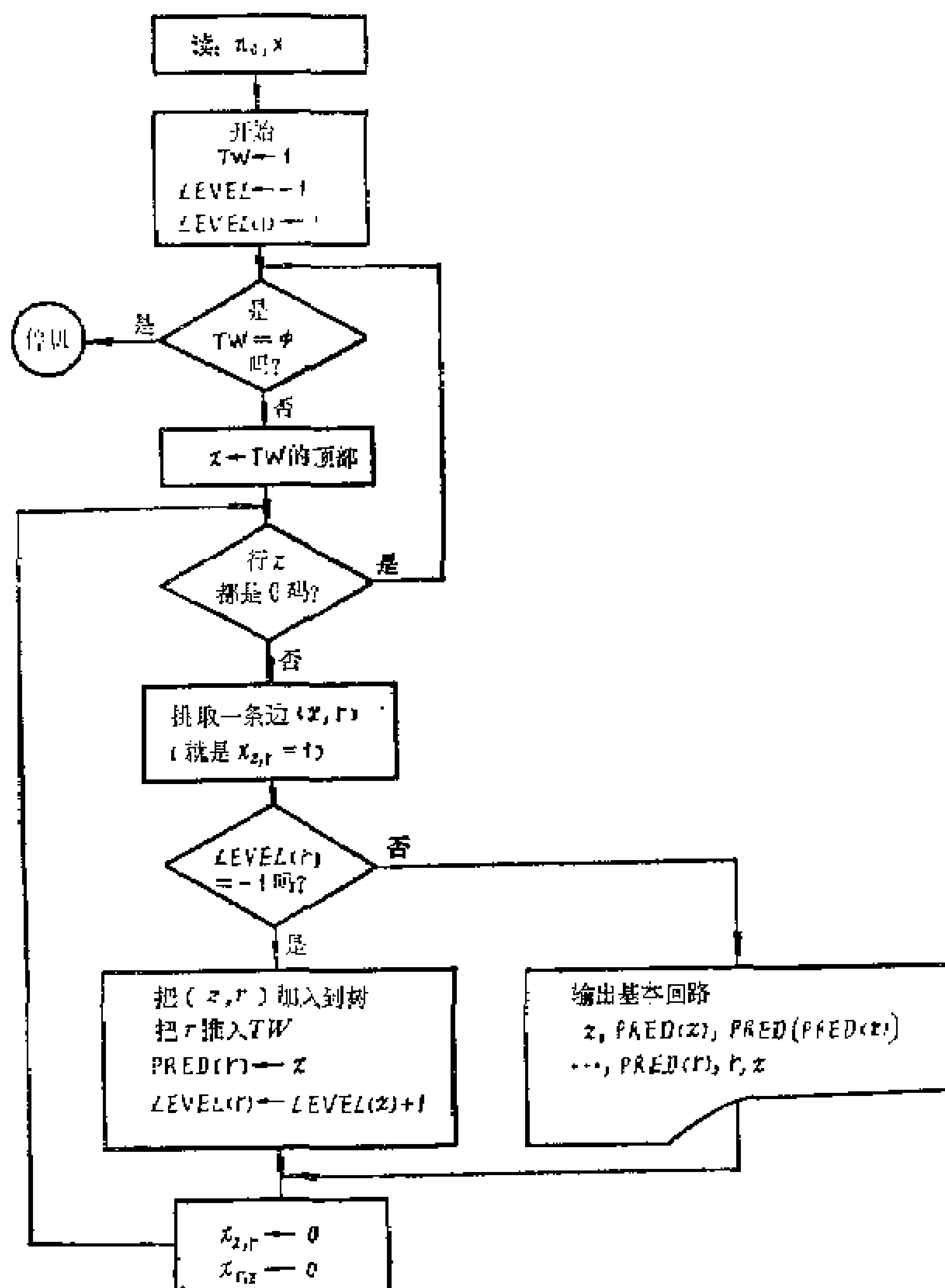


图 11—5

是因为回路的一个线性组合可以是一个回路，或是若干回路的边不共用并集，而回路的边不共用并集中包括其它回路。可是这样一种算法的效率是很差的。从N个基本回路一定会构成2^N-N-1个线性组合，其中每一个组合必须成对的与每一个其它组合相比较来测试它的内容。这是很费时的，所以要找另一个途径。

为了减少存储量和比较次数，人们可能会尝试在回路的每一个边不共用并集一产生后立

即抛弃它。可是这个途径是有缺点的，因为我们也许会发现几个抛弃掉的回路组的组合却是一个真正回路，并且由此使以后会产生另一个回路。对于这个问题，曾有人研究过并提出若干新的算法，但是都不够理想或有错误。所以如何找出一种算法来鉴别一个图中的所有回路仍是一个未解决的问题。

算法四：截割点和可分性算法

知道一个图是连通图之后，下个最可能问的问题是：图G是否是分离的？就是问G中是否有一二个截割点？如果答案是“是”的，人们会想找出G中的截割点和它的部分。

在第四章第六节中已经指出过，在通信网路脆弱性的研究中，截割点是很重要的。另外，这个算法也可以作为别的算法，如平面性和同构性算法的一个子程序。

初步简化：在这个算法中，象在其它算法中一样，是值得先做一些初步简化工作的。如果给出的图有可能分离的话，我们可以应用连通度和部分算法，并把每一部分作为一个连通图。把一个分离图的所有部分牵连在一起处理，这对于计算机的存储量和执行时间是一种浪费。同样，如果不是一个简单图，我们可以立即抛弃所有那些自环和平行边，因为它们的存在与否对于图的可分性是没有影响的。其次，如果图有悬挂点，我们可以重复地删除那些悬挂点来修整那图。在修整过程中，我们一定要留意，每一个邻接悬挂的那个点，除非那图只有一条边，否则就是一个截割点。这些简化手续通常将大体上减少原图的大小。当然，初步简化手续只在确实能节省计算机的操作时间时才进行的。

测试一个图的可分性的一个直接简单的方法是依次移去每一个点（即从它的邻接矩阵X中删除对应的行或列）后用连通度和部分算法来测试结果图的连通度。但是这是低效率的方法，所以应该另找途径。

从第四章中，可以推论出这样一个结果，当且仅当在一图中至少有一个回路包括两条边x和y时，这两条边是在同一个部分中。初看起来，要用同一个部分的这个特征，我们就得产生所有回路。这很明显是一件浪费时间和涉及存储量的事情。可是，下列两个辅助定理告诉我们只要产生一个基本回路就够了。

辅助定理11—1：一个图中两个基本回路的非空交集总是一条路径。

证：关于某个特定的生成树T，设e_1和e_2分别为组成基本回路C_1和C_2的两条弦。那么，如果$C_1\cap C_2$包含着两条边x和y它们与$C_1\cap C_2$不相连通度，在x和y之间在C_1中有一条路径P_1（就是，在边x的端点之一与边y的端点之一之间的一条路径）；这条路径不含有弦e_1。同样，在x和y之间在C_2中有一条路径P_2，这条路径不含有弦e_2。那么子图

$$P_1\cup P_2\cup(x,y)$$

就含有一个回路，它不含任何一条弦。这是不可能的。这就证明了这条辅助定理。

辅助定理11—2：在一个图G中，如果边a和b属于一个基本回路C_i，边b和c属于另一个基本回路C_j，而$a\notin C_j$，$c\notin C_i$时，那么在G中就有某一个回路Γ，a和c两条边都在Γ中。

证：由辅助定理11—1可知，因为$C_i\cap C_j$是条路径，它含有b而不含有a或c。因$C_i\oplus C_j$是一个回路（不是回路的一个边不共同并集），它含有a和c。▮

算法的说明：如果我们一个一个地产生基本回路，并且正当每一个基本回路产生时，我们把所有的边用下列步骤同样地标以数字或重新标数〔这个标数法可以用邻接矩阵来做的，而把所标数字就写在上面。在点i和j之间的边标以$q(q=2,3,\cdots,n_e-n_v+1)$。只要把$x_{ij}$和$x_{ji}$（如图上三角就是$x_{ij}$）代以$q$。仍旧是1的元素值对应着未标数边。其它边将标以2，3，…等等。〕我们会鉴定图中各个部分。

第一个基本回路的各边标以2。当找到第二个基本回路时，它的各边或者尚未标数，或者有些边已标以2。在前者的情况时，那么第二个基本回路的各边都标以3，而在后者的情况时，标以2。当我们进行到第m（$1\leqslant m\leqslant n_e-n_v+1$）个基本回路时，我们可能发现下列三种情况之一：

（一）如果第m个基本回路的各边都没有标过，则各边标以一个新的整数$g+1$。

（二）如果第m个基本回路中有些边已经标以u，而其它边尚未标数的话，那么未标各边也标以u。

（三）假使第m个基本回路中有些边已标以u，有些边已标以v，有些边已标以w，…，等等，而有些边尚未标数，设$u<v<w<\cdots$，那么把G中所有标以u，v，w，…的各边重标以u，并且把第m个基本回路中尚未标数的各边也标以u。

当这个过程完毕，在产生了各基本回路和每个回路的各边标数后，就可完成下列工作：（1）属于一个回路的每条边已经标以数字。而且，当且仅当任何两条边同在某一回路（不必是基本回路）时，这两条边有相同数字。也就是可以确定每一边组有相同数字的成为一个部分。（2）判断一个图的部分多于1的，这个图是可分的。任何一个点关联到不同标数边的是一个截割点。一条没有任何数字的边是一条桥接边。

在这个算法中，一条属于一个回路的边会更换数字好多次。这明显是低效率的渊源。所以有人建议用下列办法减少重新标数。在每一个基本回路产生出来时，各边不是立即加以重新标数，而要等到在基本回路算法中一个点z完全检查过之后，并把通过z所产生的各基本回路的各边指定同样的数字。这样标数次数就只有n_v次而不是N次了。此外，图中的各边不必都标数的。关于这些，这里不再讨论了。有兴趣的读者可以查阅文献。

算法五：有向回路算法

关于一个有向图的最重要的问题之一是它的有向回路。有向回路的重要性在第七章中已经讲到。与无向图不一样，现在还没有方法来得到有向图的一个基础组，靠了这个基础组的线性组合来得到有向图中每一个有向回路。因此，为了得到一个有向图的所有回路，算法3是没有什么帮助的，我们必须各自地产生每一个有向回路。为此，除非我们已经知道某条边不属于有向回路的，否则我们必须检查每条边许多次。

初步简化：在大多数情况下，虽然不是必要的，但是预先对一给定的有向图进行下列两步简化是有好处的。第一，如果一个有向图可能是分离的，就先用连通度和部分算法（为了是有向图，要稍加修改）去检查它的连通部分，然后每次检查一个部分。第二，相继删除所有零入次数或零出次数的各点连同关联到它们的各边，很明显，这样一个点不会包括在任何有向回路之中。因为这些点对应着邻接矩阵X中整行〔对于$d^+(v)=0$〕或整列〔对于$d^-(v)=0$〕为零，所以是容易鉴别的。例如图7—4中所示的有向图，点3和点6的入次数$d^-(3)=d^-(6)=0$，而点4的出次数$d^+(4)=0$，所以这3个点连同关联到这些点的边a，b，c，e，h，可以删除。结果这个有向图就只剩了3条边d，f和g了。在另一方面，这种简化法并不能简化如图7—23那样的有向图。

算法说明：这个算法在一个给出的有向图G中尽是搜寻所有有向回路。照例，G中各点标以整数1，2，…，n_v作为点名。算法从点p_1开始构成一个有向路径$P=(p_1,p_2,\cdots,p_k)$，一直到点p_k为止，即没有可采用的，满足一定条件的点来再延长这个有向路径了。在点p_k，算法要检查从p_k到p_1是否有一条有向边。如果有这样一条有向边，那就找到了一个有向回路（p_1，p_2，…，p_k，p_1），及时记录下来。如果在有向图中没有这样一条边，我们就后移一

个点到p_{k-1}，并且试试看从点p_{k-1}是否沿着另一条边延伸这个回路不管有否找到一个有向回路，算法禁止点p_k再作为p_{k-1}的下一次延伸，以免走向同一条路径。

找寻有向回路并且后移一个点的过程继续进行，一直要到最后回到点p_1为止。这样由p起的所有有向路径都检查过了，并且把所有有向回路都记录下来了。从次一个点开始，重复整个过程。这种重复工作由点$p_1 = 1$开始，到$p_1 = n_v$为止。

在尽量搜寻有向路径的时候，我们一定要注意下列事项：

（一）在延伸每一条有向路径时，必须避免围绕着一个有向回路兜圈子。凡是一个点已经包括在某一条有向路径中的，千万不再用它来延伸它的路径。坚持这个原则的，就可避免绕圈转。

（二）必须避免q个点q次产生一个有向回路，即在回路中的每一个点只用一次。如果路径由点p_1开始，坚持不采用点$i \leqslant p$，来延伸路径。这条规则保证找寻某一有向回路时，只从路径开始处的最小数点着手。

有向回路算法的执行步骤　　表11—2

路径P p_1	p_2	p_3	p_4	p_5	对P的作用
1	0	0	0	0	
1	2	0	0	0	
1	2	4	0	0	
1	2	4	3	0	记录回路；置$h_{4,3} \leftarrow 1$
1	2	4	0	0	
1	2	4	5	0	无回路；置$h_{4,5} \leftarrow 1$
1	2	4	0	0	无回路；$h_{4,1} \leftarrow h_{4,3} \leftarrow h_{4,4} \leftarrow h_{4,5} \leftarrow 0$；$h_{2,4} \leftarrow 1$
1	2	0	0	0	
1	2	5	0	0	无回路；$h_{2,5} \leftarrow 1$
1	2	0	0	0	无回路；$h_{2,1} \leftarrow h_{2,2} \leftarrow h_{2,3} \leftarrow h_{2,4} \leftarrow h_{2,5} \leftarrow 0$；$h_{1,2} \leftarrow 1$
1	0	0	0	0	无回路；$p_1 \leftarrow p_1 + 1$；$H \leftarrow 0$
2	0	0	0	0	
2	4	0	0	0	
2	4	3	0	0	记录回路；$h_{4,3} \leftarrow 1$
2	4	0	0	0	
2	4	5	0	0	记录回路；$h_{4,5} \leftarrow 1$
2	4	0	0	0	无回路；$h_{4,1} \leftarrow h_{4,2} \leftarrow h_{4,3} \leftarrow h_{4,4} \leftarrow h_{4,5} \leftarrow 0$；$h_{2,4} \leftarrow 1$
2	0	0	0	0	
2	5	0	0	0	记录回路；$h_{2,5} \leftarrow 1$
2	0	0	0	0	无回路；$p_1 \leftarrow p_1 + 1$；$H \leftarrow 0$
3	0	0	0	0	记录回路；$p_1 \leftarrow p_1 + 1$；$H \leftarrow 0$
4	0	0	0	0	
4	5	0	0	0	无回路；$h_{4,5} \leftarrow 1$
4	0	0	0	0	无回路；$p_1 \leftarrow p_1 + 1$；$H \leftarrow 0$
5	0	0	0	0	停　止

（三）在路径延伸时，同一路径一定只能考虑一次。在$n_v \times n_v$矩阵$H = [h_{ij}]$中准备一个最新禁用点表可以避免再用在行i的那些1的元素对应着由点i禁用这些点（就是，如果从点i禁止延伸到点j，置$h_{ij} \leftarrow 1$）。一个零值的元素表示这个点不禁用（就是，如果$h_{ij} = 0$，那么从点i可以延伸到点j）。每次一个新点选作为开始点时，矩阵H重新置0。

有向图用它的邻接矩阵输入到计算机。各点照例用整数1，2，…，n_v标号。所考虑的

有向路径用一个n_0阶线性数组来代表。

$$P=(p_1,p_2,\cdots,p_{k-1},p_k,0,0,\cdots,0,0)$$

每一条路径的第一个点为p_1，最后一个点为p_k。

例11—3　当图11—6中的有向图应用这个算法时，执行步骤如表11—2所示。

这种算法的框图如图11—7所示。

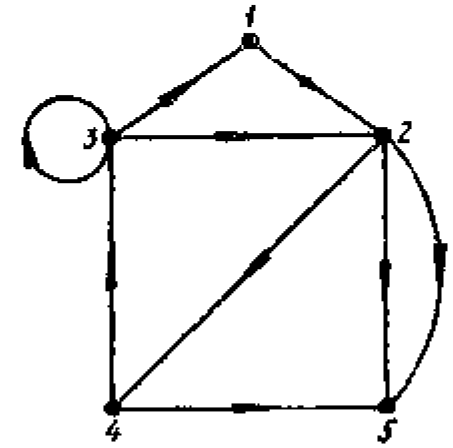


图　11—6

这种算法其实只是一个系统的彻底的搜寻有向回路的方法。在例11—3中可以看出，同一有向路径走过好几次。即使是一个有向回路，在它轮到认为“是”之前，常常是检查几次，抛弃几次。因此这种算法是很慢的。很有改善的余地。

这种算法可以很容易地加以修改以产生所有的有向汉密尔顿回路。

一个有20个点，55条边和434个有向回路的随便的有向图在IBM-7044机上约费17秒钟。这就表明这种牵涉到系统的彻底搜寻方法，很据执行时间来讲是效率很差的。

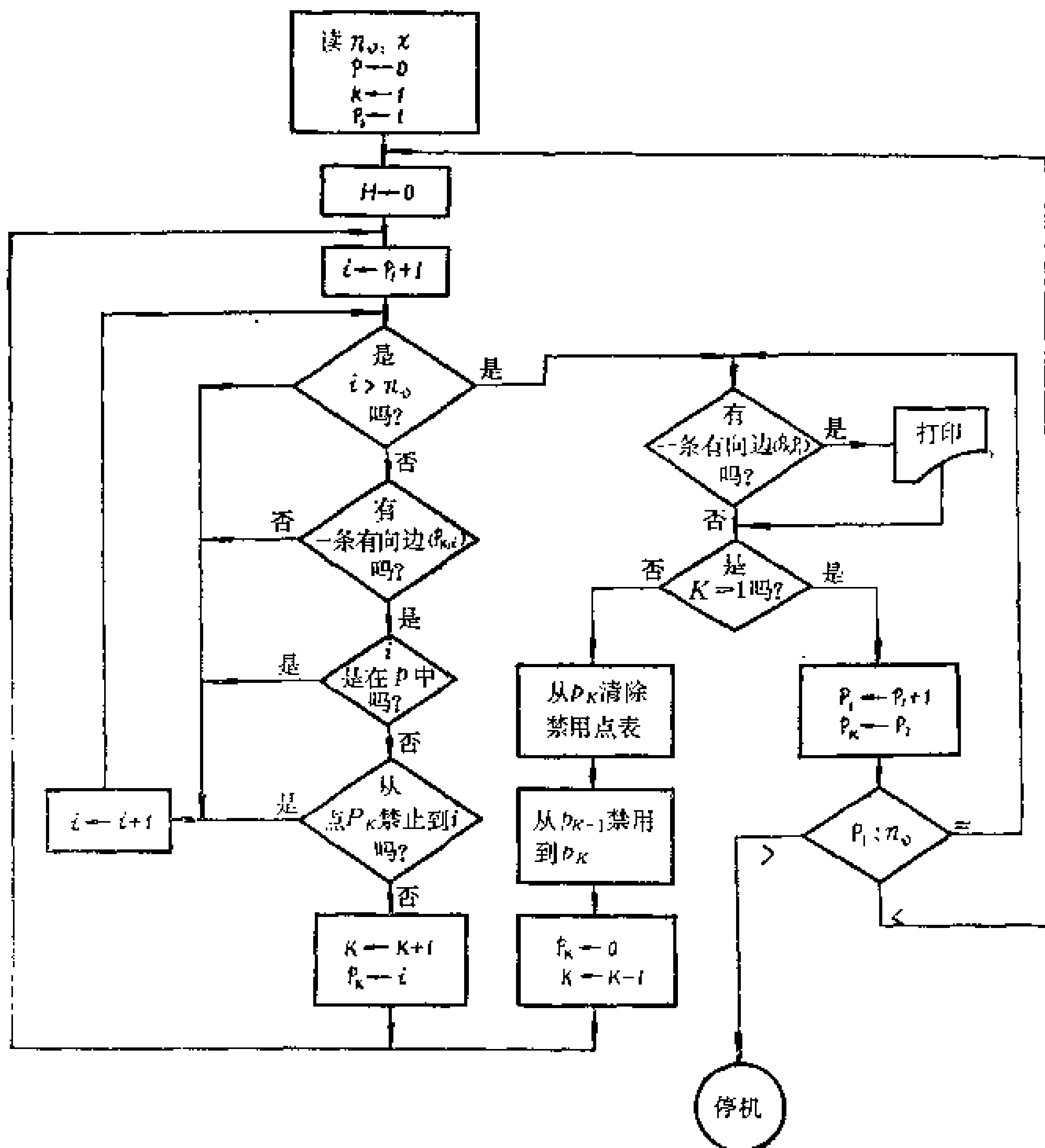


图　11—7

第六节　最短路径算法

很多的优化问题在数学上讲，是相当于找寻一个图中的最短路径。因此，在图论中，最短路径算法比其它算法研究得更为透彻。到1974年为止，大约有100多篇论文和几十种算法已经提出来了。有些算法比其它的好一些，有些更适用于某种特殊结构的图，有些只是前一些算法的小变化。

最短路径问题有不同类型，最常见到的有下列五种，其中前三种将在本节中解决：

（一）在两个规定点之间的最短路径。

（二）在各对点之间的最短路径。

（三）从某一规定点到其它所有点之间的最短路径。

（四）从某些规定点通过某些规定点之间的最短路径。

（五）第二、第三短等最短路径。

在最坏情况下，第一种问题成为第二种。因为我们即将看到，在找寻某两个规定点之间的最短路径的过程中，我们可能要确定到所有其它各点之间的最短路径。让我们先处理第一种类型的问题。

算法一：从一个规定点到另一个规定点之间最短路径的算法

从一个规定点 s 到另一个规定点 t 之间找寻最短路径的问题可以陈述如下：

这个算法原来是用来处理一个简单的加权有向图G（如果给定的有向图不是简单的，我们可以抛弃那些自环，并把平行边组代以其中最短的，或最低权边）的。同样，这个算法不一定限于处理有向图，对于一个无向图，也可适用，只是$d_{ij}=d_{ji}$，并且每一条无向边实际上可用同权的两条相反的有向边来代替。如果图是无权的，可以假定 $d_{ij}=1$，这样邻接矩阵就变成了距离矩阵。一个有n_o个点的简单加权图G可用一个$n_o\times n_o$矩阵D来表示：

$$D=[d_{ij}] \qquad (11-5)$$

式中　$d_{ii}=0$

d_{ij}＝由点i到j之间有向边的距离（或段数或权），$d_{ij}\geqslant 0$；

$d_{ij}=\infty$，如果由点i到点j之间没有边（在执行一个程序时，∞用一个大数，比如说999999来代替）。

一般，$d_{ij}\neq d_{ji}$，所以不需满足三角不等式；就是说，$d_{ij}+d_{jk}$可能小于d_{ik}。事实上，如果每一个 i，j 和 k 能满足三角不等式的话，因为直通边（x，y）就会是从点 x 到点 y 之间的最短路径，问题就简单了。一条有向路径P的距离是P中各边的距离之和。问题就在于找寻最短可能路径，和从始点 s 到终点 t 的距离。

在一对规定点之间最短路径的算法有好几种，下面介绍的算法可能是效果最好的。

算法说明：这个算法要把给出的有向图的各点标以数字。在算法的每一阶段，有些点是固定标数，而其它一些点是临时标数的。算法开始时，始点 s 指定一个固定数字 0，而其余 n_o-1 个点各暂时指定一个临时标数∞。从此之后，在每次迭代中，另一个点按照下列规则得到一个固定标数：

（一）还没有固定标数的每一个点 j 得到一个临时标数，它的值为

$$\min\{(j\text{的旧标数}),\ [(i\text{的旧标数})+d_{ij}]\} \qquad (11-6)$$

即选取（j 的旧标数）和[（i 的旧标数）$+d_{ij}$]中的较小数，其中 i 是前一次迭代中最新固

定的标号，d_{ij}是点i和j之间的直接距离。如点i和j之间没有一条边相连，则$d_{ij}=\infty$。

（二）找出所有临时标数中的最小数，这就变成这个对应点的固定标数。当有一对同样标数点时，两者中任选一个点，给以固定标数。

这两步交替重复进行，一直要到终点 t 得到一个固定标数为止。

第一个固定标数的点离开点 s 的距离为 0，第二个得到固定标数的点（即余留下的n_v-1个点中之一）是最近点 s 的点。从余留下来的n_v-2个点中下一个得到固定标数的点是第二个最接近 s 的点。余类推。每一个点的固定标数就是这个点与点 s 之间的最短距离。

例11—4　有一个有向图如图11—8所示。

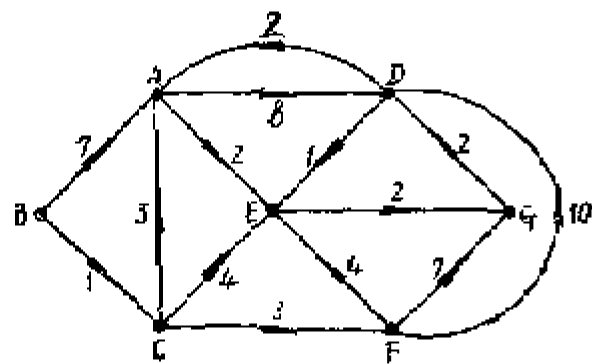


图　11—8

找寻由点B到点G的最短距离。我们在进行解决这个问题的过程中，找出一表如下来表示各点临时和固定标数，其中固定标数用一方框围着，而最新指定的固定标数用一个记号✓表示。标数进行如下：

$A\ B\ C\ D\ E\ F\ G$

∞[0]

7 [0]✓∞∞∞∞∞：始点B标以 0

7 [0] 1 ∞∞∞∞：所有B的后继点标以数

7 [0] [1]✓∞∞∞∞：最小标数变成固定标数

4 [0] [1]∞ 5 4 ∞：所有C的后继点标以数

4 [0] [1]∞ 5 [4]✓∞：最小标数变成固定标数

4 [0] [1]14 5 [4]11：所有F的后继点标以数

[4]✓[0] [1]14 5 [4]11：最小标数变成固定标数

[4] [0] [1]12 5 [4]11：所有A的后继点标以数

[4] [0] [1]12[5]✓[4]11：最小标数变成固定标数

[4] [0] [1]12[5] [4] 7：所有E的后继点标以数

[4] [0] [1]12[5] [4] [7]✓：终点G得到固定标数

各步是容易编制程序的，只是把固定标数点区别于临时标数点的工作稍需技巧。完成这个工作的有效方法是把各点伴以下标1，2，…，n_v，并且保持一个n_v阶的二进制矢量$VECT$。当点 i 变成固定标数时，这个二进制矢量的元素 i 就由 0 改为 1。

两个规定点之间最短路径算法的框图见图11—9。

这里附有一个准备执行的$BASIC$程序以便找寻图11—8中由点A到点G之间的最短路径。在图11—8中图G的距离矩阵中∞用99来代替。

规定点之间最短路径程序

```
10 REM SHORTEST DISTANCE FROM S TO T
20 LPRINT "NO.OF VERTICES=? "
30 INPUT N
40 LPRINT "S=STARTING VERTEX=? "
50 INPUT S
```

```
60 LPRINT "T=TERMINAL VERTEX=? "
70 INPUT T
80 FOR I=1 TO N
90 FOR J=1 TO N
100 READ D(I, J)
110 NEXT J
120 NEXT I
130 FOR I=1 TO N
140 FOR I=1 TO N
150 L(I)=99
160 V(I)=0
170 NEXT
180 L(S)=0
190 V(S)=1
200 I=S
210 M=99
220 FOR J=1 TO N
230 IF V(J)=1 THEN 300
240 REM Z=PERMANENT LABEL OF VERTEX
250 Z=D(I, J)+L(I)
260 IF Z<L(J) THEN L(J)=Z
270 IF L(J)>M THEN 300
280 M=L(J)
290 P=I
300 NEXTJ
310 V(P)=1
320 IF P=T THEN 350
330 I=P
340 GOTO 210
350 LPRINT "SHORTEST DISTANCE FROM S TO T=L(T)"
360 DATA 99,99,99,8,2,99,99,7,99,1,99,99,99,99,3,99,99,99,4,3,99,2,99,99,99,1,99, 99,
    99,99,99,99,99,99,2,99,99,99,10,4,99,7,99,99,99,2,99,99,99
370 END
```

上述算法实际上不是列出从始点到终点的最短路径，而只给出最短距离。最短路径可用下法构成，即从终点倒退到那个前趋点，使两点的差数正好是连接边的段数。如有同样标数点时，表示最短路径不止一条。另一种可采取的办法是把已经固定标数的各点记录下来，以确定最短路径。这个记录可以用段数为n,的另一个线性数组的方式保存下来。每当有一个新的而定数字指定给点 j 时，从 j 直接可达的那个点总是记载在这个数组的位置 j 中。

这个算法的注意事项：

（一）在这个算法中，如果我们进行到每一个点都得到一个固定标数，而不是停止在终点 t 的固定标数上时，那么这个算法就是找寻由某一规定点到其它所有点之间最短路径的算法。

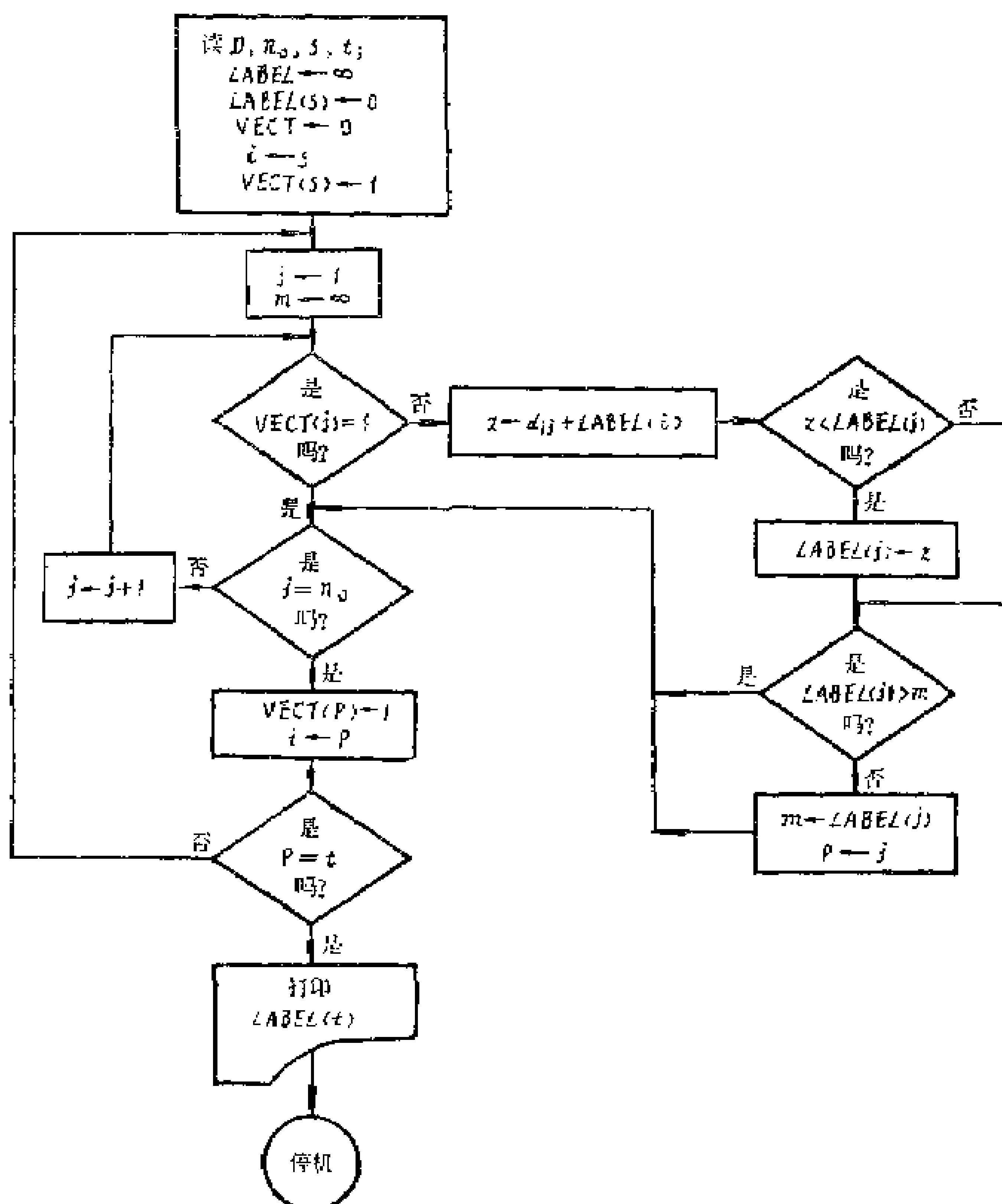


图 11—9

(二) 如果我们取从始点 s 到每一个由 s 可以达到的其它点的最短路径，那么这些路径的并集将是一个生根于点 s 的外向树 T。在 T 中从 s 起的每一条路径是有向图（也许有些情况是无向图）中唯一最短路径。这样的一棵树叫作最短距离外向树。如果是无向图，就是最短距离树，它与最短生成树有别。在进行算法 6 时，如果标数继续到每点都得到一个固定标数，并且如果每次只有一个点给予固定标数时，相应的边添加到外向树，这棵外向树就可以作为从一个规定点到另一个规定点之间最短路径的算法的副产品来构成。例如，图11—8 的最短距离外向树就如图11—10所示。

(三) 在这个算法中，当更多的点得到固定标数后，需要用以改变临时标数的加法和比较就逐渐减少。在每一个点得到固定标数的情况下，我们需要用$n_v(n_v-1)/2$次加法和 $2n_v(n_v-1)$次比较。所以计算时间是与n_v^2成正比的。

(四) 注意当有向图的点数为n_v时，计算时间是与有向图可能有的边数n_e无关的。这是因为有意假定这是一个完全有向图。如有缺边，只要给与一个很大数。下列典型数据也可证

实这一点：在IBM-7044机上，对于一个有80个点和3200条边的任意有向图，用*FORTRAN*程序化费 36/60 秒钟去找寻由一个规定点到所有其它点的最短距离。另一个有80个点和1000条边的任意图，也化费36/60秒钟作相同计算。

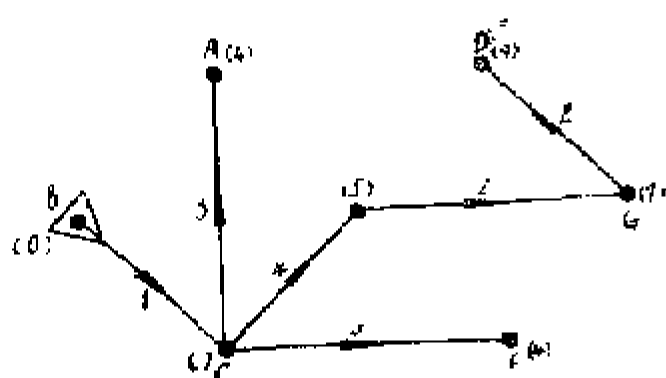


图 11—10

（五）如果有向图是稀疏的，就是边数n_e小于$n_v(n_v-1)$很多的，就有可能减少计算时间。这可以结合另一种测试法去改变那些只是新近固定标数的后继点的临时标数来达到的，当然，这要花费些测试时间来换取由于测试结果而节省下来的时间。另一种减少稀疏图的计算时间的方法将在下一个算法的注意事项第（三）中讲到。

（六）如果给出的有向图G不是加权的，那么G中每一条边的权可以作为 1，这样矩阵D就与邻接矩阵相同了。那么问题就简单些，我们只要进行逻辑操作而不是实在的算术操作了。

（七）我们假定了距离d_{ij}都是些正数。如果有些距离是负的，这个算法就不能工作了（在网络中的正距离可能代表盈，负距离代表亏）。这个算法所以不能工作的理由是因为一个点的标数已经固定后就不能改变了。曾有人建议过用最短路径算法来解决这种问题。不过有一个条件，就是环绕着每一个有向回路的所有d_{ij}的和必须是正数。能处理负d_{ij}的计算时间是与n_v^3而不是与n_v^2成正比的。

（八）曾有人建议过，如果把最短路径算法从始端 s 和终端 t 同时进行，会改变速度。不过也有人指出这个算法的双端进行法只会改进某些类型有向图的效率。在一个有向图中，差不多所有n_v个点都必须从这一端或那一端进行固定标数时，双端进行法实际上比上面讲过的单端进行法的效率还低些。

算法二：每对点之间最短路径的算法

有时候，人们对于找寻一个有向图中所有$n_v(n_v-1)$个有序点对（或在一个无向图中$n_v(n_v-1)/2$个无序点对）之间的最短路径是感兴趣的。如果我们还是用从一个规定点到另一个规定点之间最短路径的算法来达到这个目的，计算时间会与n_v^4成正比的。现在有几种算法可以用来更好地完成这个任务，其中有两种可以算是最好的，效率相同。两者的计算时间与n_v^3成正比的。我们将介绍其中的一种。

算法说明：当算法在进行中认为有利时，就在路径中插入一个或几个点。

由直接距离的$n_v\times n_v$，矩阵$D=[d_{ij}]$开始，n_v个不同矩阵D_1，D_2，…，D_{n_v}相继构成。矩阵D_k，$1\leqslant k\leqslant n_v$，可以看作是那样的一个矩阵，它的$(i,j)$处元素值给出由$i$到$j$之间所有有向路径中最短有向路径的段数，点 1，2，…，k可作为中间点。矩阵$D_k=[d_{ij}^{(k)}]$是依照下列规则从D_{k-1}构成的：

$$d_{ij}^{(k)}=\min\{d_{ij}^{(k-1)},(d_{ik}^{(k-1)}+d_{kj}^{(k-1)})\} \qquad (11—7)$$

其中$k=1$，2，…，n_v，$d_{ij}^{(0)}=d_{ij}$，就是说，在迭代中，如果$d_{ij}>d_{i1}+d_{1j}$，就把点 1 插入到由点 i 到 j 的路径中。在迭代 2 中，把点 2 插入，等等。

例11—5　假定由点 7 到点 3 的最短有向路径为741953，就有下列替换：

迭代 1：$(d_{41}^{(0)}+d_{13}^{(0)})$替换$d_{43}^{(0)}$

迭代 4：$(d_{74}^{(3)}+d_{43}^{(3)})$替换$d_{73}^{(3)}$

迭代 5：$(d_{85}^{(4)}+d_{53}^{(4)})$替换$d_{83}^{(4)}$

迭代 9：$(d_{78}^{(8)}+d_{83}^{(8)})$替换$d_{73}^{(8)}$

一旦在$d_{73}^{(8)}$中得到最短距离时，在其后运算中，这个元素值就不再变更了。

这个算法的框图见图11—11。

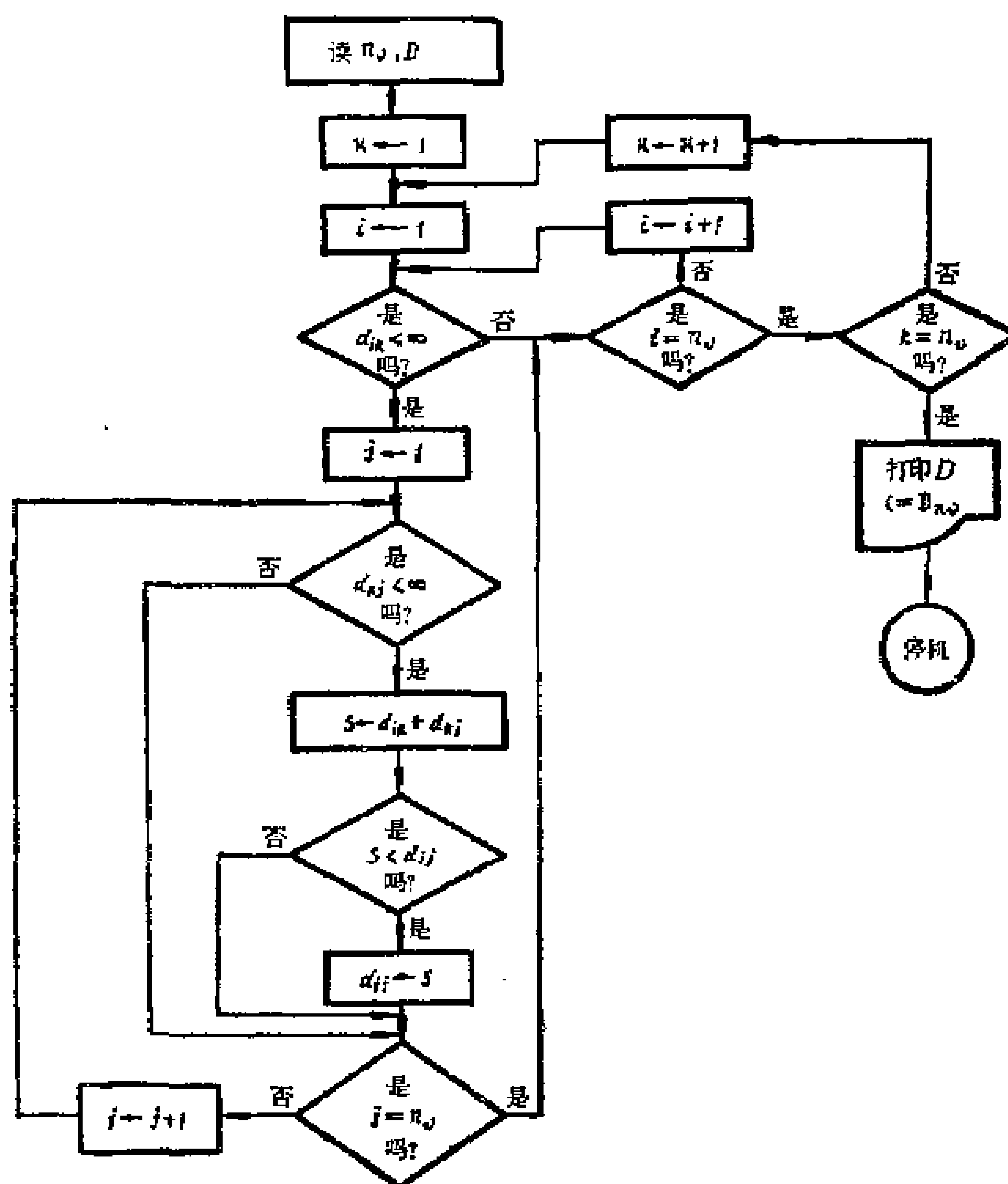


图 11—11

这里附有一个准备执行的$BASIC$程序以便找寻图11—8的有向图中每对边之间的最短路径。这个程序的输入数据与从一个规定点到另一个规定点之间最短路径的算法中所附程序的数据相同。

每对点之间的最短路径程序

```
10  REM SHORTEST PATH BETWEEN EVERY VERTEX-PAIR
20  READ N
30  FOR I=1  TO N
40  FOR J=1  TO N
50  READ D(I,J)
60  NEXT J
70  NEXT I
```

```
80 FOR K = 1 TO N
90 FOR I = 1 TO N
100 IF D(I,K) < 99 THEN 120
110 GOTO 180
120 FOR J = 1 TO N
130 IF D(K,J) < 99 THEN 150
140 GOTO 170
150 S = D(I,K) + D(K,J)
160 IF S < D(I,J) THEN D(I,J) = S
170 NEXT J
180 NEXT I
190 NEXT K
200 FOR I = 1 TO N
210 FOR J = 1 TO N
220 D(I,I) = 0
230 LPRINT D(I,J);
240 NEXT J
250 LPRINT
260 NEXT I
270 DATA 7,99,99,99,8,2,99,99,7,99,1,99,99,99,99,3,99,99,99,4,3,99,2,99,99,99,1,99,99,
    99,99,99,99,99,99,2,99,99,99,10,4,99,7,99,99,99,2,99,99,99
280 END
```

讲到这里为止的算法实际上并不是列出路径，而只是给出最短距离。要得到路径，因为现在要有$n_v(n_v-1)$条路径而不只是一条，所以要比从一个规定点到另一个规定点之间最短路径的算法较复杂些。在每个最短路径得到中间点的一个有效方法是组成一个矩阵$Z=[z_{ij}]$，这种矩阵叫作最优策略矩阵，元素值z_{ij}是由 i 到 j 沿最短路径中从点 i 起所遇到的第一点。最优策略矩阵 Z 可用下法组成：

开始，我们置

$$z_{ij}=j,\quad 如 d_{ij}\neq\infty$$
$$z_{ij}=0,\quad 如 d_{ij}=\infty$$

在第 k 次迭代中，如果点 k 插入 i 和 j 之间，元素值z_{ij}就由z_{ik}的当时值所替换。这对于所有 i 和 j 都是这样。在每次迭代 k 中，其中 $k=1,\ 2,\ \cdots,\ n_v$，Z 矩阵就更新。结果，从 i 到 j 的最短路径（$i,\ v_1,\ v_2,\ \cdots,\ v_q,\ j$）就从矩阵 Z 中各点数的序列中如下导出：

$$v_1=z_{ij}$$
$$v_2=z_{v_1,\ j}$$
$$v_3=z_{v_2,\ j}$$
$$\vdots$$
$$j=z_{v_q,\ j}$$

这个算法的注意事项是：

（一）注意为了计算目的，我们只需要一个$n_v\times n_v$矩阵的存储量。其它所组成的矩阵可以在这个矩阵上修改。

（二）为了估计时间，我们得考虑要相继组成n_v个矩阵D_1，D_2，$\cdots$，D_{n_v}。关于每个矩

阵D。因为在式（11—7）中，$i \neq j$，$i \neq k, j \neq k$（虽然为了简单起见，我们在框图中没有利用这细小的节省）计算的元素数为$(n_v - 1)(n_v - 2)$。因此执行时间是与$n_v(n_v - 1)(n_v - 2) \simeq n_v^3$成正比的。

（三）当式（11—7）的$d_{ik}^{(K-1)} = \infty$时，这总是可能用额外的测试来避免（$n_v - 1$）次加法和比较的。这正象在从一个规定点到另一个规定点之间最短路径的算法中的折衷办法。但是因为现在这个算法的执行时间是与n_v^3成正比、所以这是值得对差不多每一个有向图进行这个额外测试的。注意在框图中，这个测试没有包括进去。

（四）如果图是稀疏的，就是边数远少于$n_v(n_v - 1)$的话，这就可能利用这种特殊结构，把图分解开来以减少工作。在每一个子图中找出最短路径。作为一个极端的例子，假定一个有n_v个点的有向图是由各有$n_v/2$个点的两个有向图所合成的，那么计算时间就由n_v^3减少到$2(n_v/2)^3$，即减少了75%。

（五）象在从一个规定点到另一个规定点之间最短路径的算法中一样，如果有向图是无权的，就是$D = X$，计算时间就可以由于用逻辑操作代替了算术操作而减少了。

一个有向图的传递闭包：设G是一个简单的有n_v个点的有向图，让我们构造另一个简单的有n_v个点的有向图，象下面那样把一些边加到G中去。当且仅当在G中由i到j有一条有向路径（有任何段数2，3，…，$n_v - 1$）时，由点i到j加一条边（i，j）。这个有向图H叫作是G的传递闭包。换句话说，

$$X(H) = R_{(G)}$$

其中$X(H)$是H的邻接矩阵，$R(G)$是G的可达性矩阵。

一个给出的有向图G的传递闭包是可以对邻接矩阵$X(G)$应用本算法来得到的，就是置$D \leftarrow X$。用这个方法来得到传递闭包的计算时间是与n_v^3成正比的。

最长路径分析：有时候，特别是在任务两络中的关键路径法（将在第十三章第四节中讲）中，人们需要的是从某一规定点到所有其它点的最长路径而不是最短路径。人们也许会希望用最大化方法来模拟最短路径算法中所用的最小化方法会提供所需要的路径。但是，对于一个任意的有向图，这是不行的。因为在最大化方法中，人们会围绕着一个有向回路转，段数或距离会弄得任意大。另一个困难是在最短路径问题中，如果（s，t，u，…，f）是从s到t的一条最短有向路径，那么子路径（t，u，…，f）是从t到f的一条最短路径。这是最短路径算法所依据的一个特性。在另一方面，如果（s，t，u，…，f）是s到f的一条最长有向路径，那么就有可能存在着从t经f到t的一条有向路径，这是比子路径（t，u，…，f）更为长些。

如果给出的有向图是非循环的（这是任务网络中的情况）上述两个困难就都不出现。所以上述从一个规定点到另一个规定点之间最短路径的算法就可以用来找寻一个非循环有向图中从一个规定点到所有其它点的最长路径。

除了在最短路径算法中所讲的3个最短路径问题外，还有几个其它最短路径问题。例如，有人可能要找寻从s到f的第二最短路径，或者有人有兴趣找寻从s经由某些规定点到f的一条最短路径。这些问题这里不再讨论了。

第七节　图的深度第一搜索法

深度第一搜索法也称回溯法。是对一个给定图的系统地穿行，使得每条边正好穿行过一

次，每一个点至少通过一次。

关于一个给出的图G，为了回答差不多任何一个重大问题，我们必须检查每条边（在这个过程中就牵连到每一个点）至少一次。例如，在宣告一个图G是分离的之前，我们一定要检查G中每条边。因为不是这样做的话，就可能发生我们所忽视了的那条边正好是证明图是连通的那条边。同样，对于可分性、平面性和相类似的问题也是一样。

在一个图中从一个点到另一个点地搜寻和扫描各边的方法有两种：（1）到一个点v时，我们就可以扫描所有关联到v的所有各边，然后进到邻点w。在点w，我们搜寻所有关联到w的各边。这个过程继续下去，一直到图中所有各边都扫描过为止。这个从一个点成扇形销开的方法称为图的广度第一搜索法。这就是用于基本回路算法的方法，也隐含于连通性和部分算法和最短路径算法中。（2）与从一个点扫描所有关联到这个点的各边正相反的另一途径是从一个点尽快的进行到一个尚未去过的邻点w，而暂时不管关联到v而可能尚未检查过的那些边。换句话说，只要可能，我们总是沿着图中一个行走通到一个新点。这种穿行图的方法称为深度第一搜索法，由于对各点编号和对各边定向的结果，这种方法对于简化图论算法是很有用的。

各点编号和各边定向：当对一个图进行深度第一搜索法，并且第一次检查到一个点v时，我们总是指定一个不同序列数$NUM(v)$给它。如果点v是穿行中所检查到的第i个点，$NUM(v)$就是i。同样，穿行图中所经过每一条边就照穿行的方向给与一个定向。所以在搜索结束时，深度第一搜索法所进行过的无向图G就变成一个有向图G了，它的各点编号下列各数1，2，…，n。深度第一搜索法算法的详细步骤见下。

深度第一搜索法算法的说明：设G是给出的无向图，用列举后继点法输入计算机。设点x是搜寻开始的规定点，$PALM$和$FRON$是G中各边所划分的两个边不共用子组。

第一步：置$v\leftarrow x$，$i\leftarrow 0$，$PALM\leftarrow\phi$，$FRON\leftarrow\phi$；

第二步：置$i\leftarrow i+1$，$NUM(v)\leftarrow i$；

第三步：找寻一条关联到v但没有穿行过的边。

1. 如果没有这样的边（就是，关联到v的每条边都已经穿行过），就进行第五步；否则将按2.那样。

2. 在点v上挑取第一条没有穿行过的边，并且通过这条边。这条边（v，w）的定向是从v到w，现在我们到达点w。

第四步：

1. 如果点w是这次搜寻中还未到达过的点（就是，如果$NUM(w)$还未规定的），在组$PALM$中加入边（v，w）。置$v\leftarrow w$，并进行第二步。

2. 如果点w是一个先前已经到达过的点（就是$NUM(w)<NUM(v)$），在组$FRON$中加入边（v，w）。进行第三步，就是回到点v。

第五步：检查组$PALM$中是否有已经穿行过的边（u，v）朝着v定向的。

1. 如果有这样一条边，退回到点u（注意u是从那里第一次到v的）。置$v\leftarrow u$，并进行第三步。

2. 如果没有这样的边（u，v）停止（我们已经通过每一条边并到过连接到x的每一个点，我们回到了根x）。

例11—6　在图11—8中，这个图G有5个点，8条边。开始点规定为x。搜寻各边的顺序见图11—12（a），根据这个穿行顺序的图见图11—12（b）。

棕榈树和叶：如果这个深度第一搜索法应用到任何一个连通的无向图G（有n_0个点和n_e条边），不难看出，等到图G中所有点都编了号1，2，…，n_0，并且所有边都定向后，这就终止了。设G是所得到的有向图。在组$PALM$中，在搜寻过程中，它的n_0-1条有向边中每一条都引向一个新点。这个子图（由边组$PALM$所确定）中的每一个点，除根x外，入次数都等于1，而x的入次数为零，所以是G中的一个外向生成树。这个外向生成树叫作棕榈树，那些不在棕榈树上的边（就是属于组$FRON$的那些边）叫作叶子或叶。对于每张叶子（c，a），因为点a已经在点c之前先到达过，所以$NUM(a)<NUM(c)$（就是根据图（b）的编号，$2<4$）。换句话说，每张叶子的定向是从高编数点指向低编数点的。

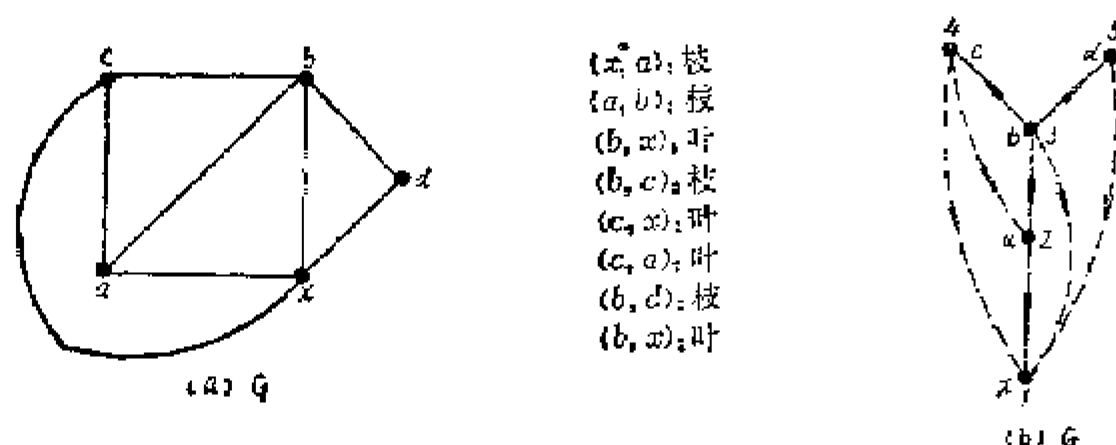


图 11—12

深度第一搜索法的本身并不能揭露一个给出的图的特性（除了告诉我们G是否是连通的）。可是，它却能把一个图的各点系统地编号，并把那些边根据刚才讨论过的性质划分成两个组$PALM$和$FRON$。这就是使深度第一搜索法成为解决数目图论问题中的一个有力工具。应用深度第一搜索法所构成的算法所能解决的一些问题如下：

（1）鉴定分离图的部分，（2）鉴定可分图的部分和截割点，（3）鉴定连通度为3或更多的最大子图，（4）平面图的平面性，（5）平面图的同构性，（6）在一个有向图中鉴定片段，即最大强连通子图。

在上述的六种应用中，文献中指出，如果图是用列举后继点法输入，因为每条边一定至少检查一次，计算时间都是与图的边数n_e成正比的。当然，这是这些算法的下限。

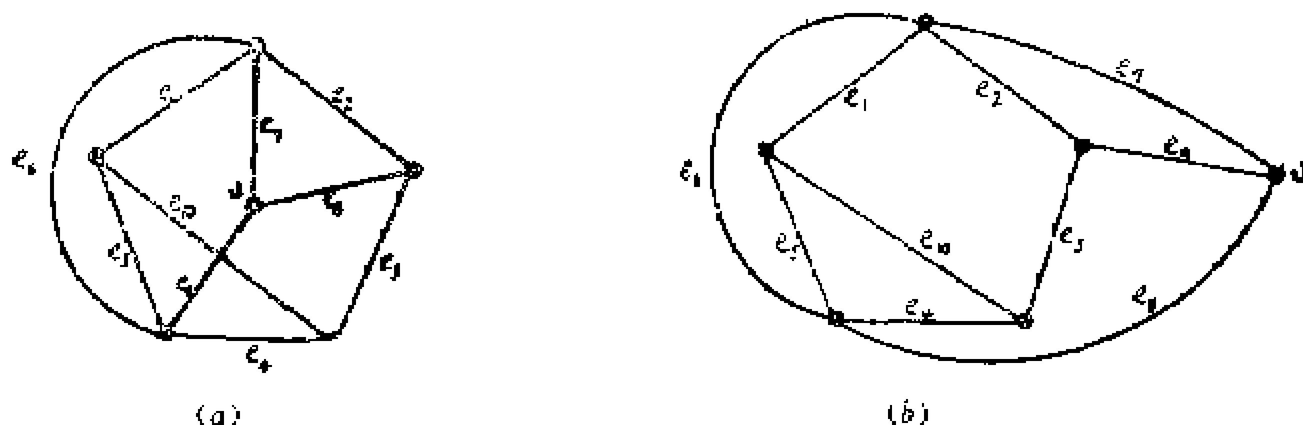
现在我们要概要地讲到平面性算法。这是用深度第一搜索法解决问题的巨大成功，它比以前的一些方法在计算时间方面有惊人的改进。现在先讲讲平面性测试问题。

确定一个给出的图是否是一个平面图是一个重要问题。在第六章中，我们讲过图的平面性的3个表征，这些表征在理论上讲是极好的，但是都不适用于计算机来测试。它们难于应用于大图；而且，如果是平面图的话，也不提供通常所需要的平面表示法。例如，如果用定理6—14来测试一个n_0个点图（$n_0>5$）的平面性，计算时间至少与n_0^6成正比的。

近年来对于平面性测试方面的算法有很多提了出来，并且设计出计算机用的程序。这些方法中的大多数是采用构图法途径，方法大约如下：先选取给出图G中的一个平面子图g（在大多数算法中，g是一个回路）。先把它映射到一个平面上，然后逐渐地把其余那些边加到g上去，不使出现交叉。如果这个再构图工作是成功的，那么图G很明显是平面的。我们就得到了G的平面表示法。如果失败的话，G是非平面的。

这样一个算法中唯一困难部分是在把哪些边加入到g的早期中，如何选择边的放置法。在早期的一次错误选择，即使这个图是平面的，也许会妨碍以后一条边的加入。

例11—7　在图11—13（a）中，如果开始的子图是回路（e_1，e_2，e_3，e_4，e_5），那么在我们加入边e_6，e_7，e_8，e_9时是不会发生任何交叉的。可是，在加入最后一条e_{10}边时却不得不发生一次交叉了。这样，我们是否就错误地认为G是非平面的呢？不是的。如果我们选择把点v不是放在回路（e_1，e_2，e_3，e_4，e_5）之中，即有限区中，而是把v放在无限区中，如图（b）所示，我们就得到了一个平面表示法。



（a）　　（b）

图　11—13

这个问题实质上是平面性测试中的框图方法，可以有不同的解决方法。

初步简化：在第六章第五节中已经讲过，一个任意图，如果遵照某些简化步骤，一般是可以缩减成小得多的图的，而所得的结果不致影响一个图的平面性（或非平面性）的。

（一）应用连通度和部分算法来检查连通度。如果这个图是分离的，每次只能检查一个部分。

（二）去除所有自环，并且把每组平行边代以一条边。

（三）合并两条串联边来删除次数为2的那些点。交替和重复应用第二步和第三步一直到图不能再简化为止。

（四）应用截割点和可分性算法将可分图划分为若干个部分，即若干个最大不可分子图。

（五）将每一部分交替应用第三步和第二步一直到不能再简化为止。

（六）这样得到的简化了的部分有n_e条边和n_v个点，按下列测试：

$$\begin{aligned} &n_v \geqslant 5 \\ &n_e \geqslant 9 \\ &n_e \leqslant 3n_v - 6 \end{aligned} \qquad (11-8)$$

如果一个图G的$n_v < 5$或$n_e < 9$，那么G是平面的；如果一个简单图的$n_e > 3n_v - 6$，那么G是非平面的。所以式（11—8）中任何一个不等式不能满足的话，我们的工作就完毕了，就可以测试第二部分了。

关于平面性的测试算法有好几种。下面所介绍的是一种效果比较好的算法。

算法说明：平面性测试算法是很复杂的，我们将只概略地说明其中的要点。为了理解主要算法，让我们把下面讲的分解法应用到一个有n_v个点和n_e条边的简单连通图G中去。

回路-路径分解法：

第一步：在G中找寻某一个回路K。置g←K。把g的各点和各边分别标以v_1，v_2，…，和e_1，e_2，…。置$i \leftarrow 1$。

第二步：如果G中有一条未标边，那么就找寻一条路径P_i，它由一个已标点开始，到另一个已标点终止，但是只包括些未标边。存储P_i。如果G中没有未标边，就进行第四步。

第三步：置$g \leftarrow g \cup P_i$。置$i \leftarrow i+1$。把g中未标边和未标点标号。返回到第二步。

第四步：停机。打印g，P_1，P_2，…，P_m。

上述方法把一个简单连通图G分解成一个回路和$m=n_e-n_v$条路径。因为回路可以看作是两个边不共用路径，所以G分解成n_e-n_v+2条路径。虽然一个图G这样地分解可能不止一种，可是G所分解成的路径数却是不变的，即等于一个回路和n_e-n_v条路径。

例11—8　在前面的图11—13中，有下列两种分解法，每种都是一个回路和4（10－6＝4）条路径：（1）（e_1，e_2，e_3，e_4，e_5），（e_6），（e_{10}），（e_7，e_8），（e_9）；（2）（e_1，e_2，e_3，e_4，e_5），（e_7，e_8），（e_9），（e_6），（e_{10}）。

在这个回路路径分解法中，我们可以把回路K映射到一个平面上，并且连续地把所产生的新路径P_1，P_2，…一个一个地加上去。加入一个新路径P_i或者把原来有的一个区划分为两个新区，或者当g是一个平面图时，$g \cup P_i$成为非平面的。这个任意加入所产生的路径的方法可能发生例11—7中所讲到的情况。为了解决这个问题，可以采用下列的两种路径之一：

（一）继续加入路径，一直到不能再加入路径为止，然后回过头来探索早期采用的另外的选择放置法。

（二）先研究应该放置的不同路径，但不急于加入到K中去，要等到找到那个区必须放入时才把一条路径加入进去，或者确信不管路径放入哪个区都无关紧要时，才把一条路径加入进去。

有的算法采用了上述的第一种途径，但有的算法采用了第二种途径，而且以为由于采用了第二种途径而使算法更为有效。后者加入路径时解决分岐的要旨在于：

假定在任何一个阶段，我们有一条路径P_i（在路径的后进先出表的顶部）它的如何放置法正在犹豫不决。设a和b是P_i的端点。图11—14中框图表示出可能出现的不同情况以及每种情况所采取的动作，这些步骤用图11—15来加以解释。

在图11—15（a）中，路径P_i在点a和b都可以自由旋转，所以这条路径或者划分路径P_j上面的区，或者划分P_j下面的区。这条路径P_i的分岐必须解决。为了解决这个分岐，在路径P_i的中间点x开始构成一条新路径P_k，这条路径P_k包括些未标边，并且在一接触到一个已标点时立即终止。

如果P_k中的x和y都在路径P_i中，如图11—5（b）所示，P_k可以在点x和y旋转。这样或者划分P_i上面的区，或者划分P_i下面的区。因此，我们不仅解决不了放置路径P_i的分岐，而且又有一条新路径P_k的分岐必须先予解决。路径P_k在存储栈中放在P_i之上，并且我们正象对待P_i那样开始解决它的分岐。

另一个可能性是P_k的终点y既不在路径P_i上，也不在P_j上，而是在另一条路径上，如图11—15（c）所示。在这种情况下，P_i不能旋转。所以P_i所划分的区是明确的。

在图11—15（d）中，如果路径P_k终于路径P_j中a和b之间的一个点。那么P_i（以及P_k）仍旧可以环绕着a和b旋转，所以分岐还是存在。可是，路径P_k划分了区$axbya$则是明确的。因此我们为了解决放置P_i的分岐，将在P_i上的一个点上产生另一条路径。

最后当路径P_k终于P_j上的一个点，但是不在a和b之间，如图11—15（e）所示，路径P_i仍能旋转。不过，不象图11—15（b）那样，路径P_k与P_i的关系不是含糊的。因此为了解决P_i的分岐，必须产生一条新路径，并且这条路径可以从路径（y，a，x，b）中任何一个点开始，那就是把路径P_i延伸到P_j上的点y。

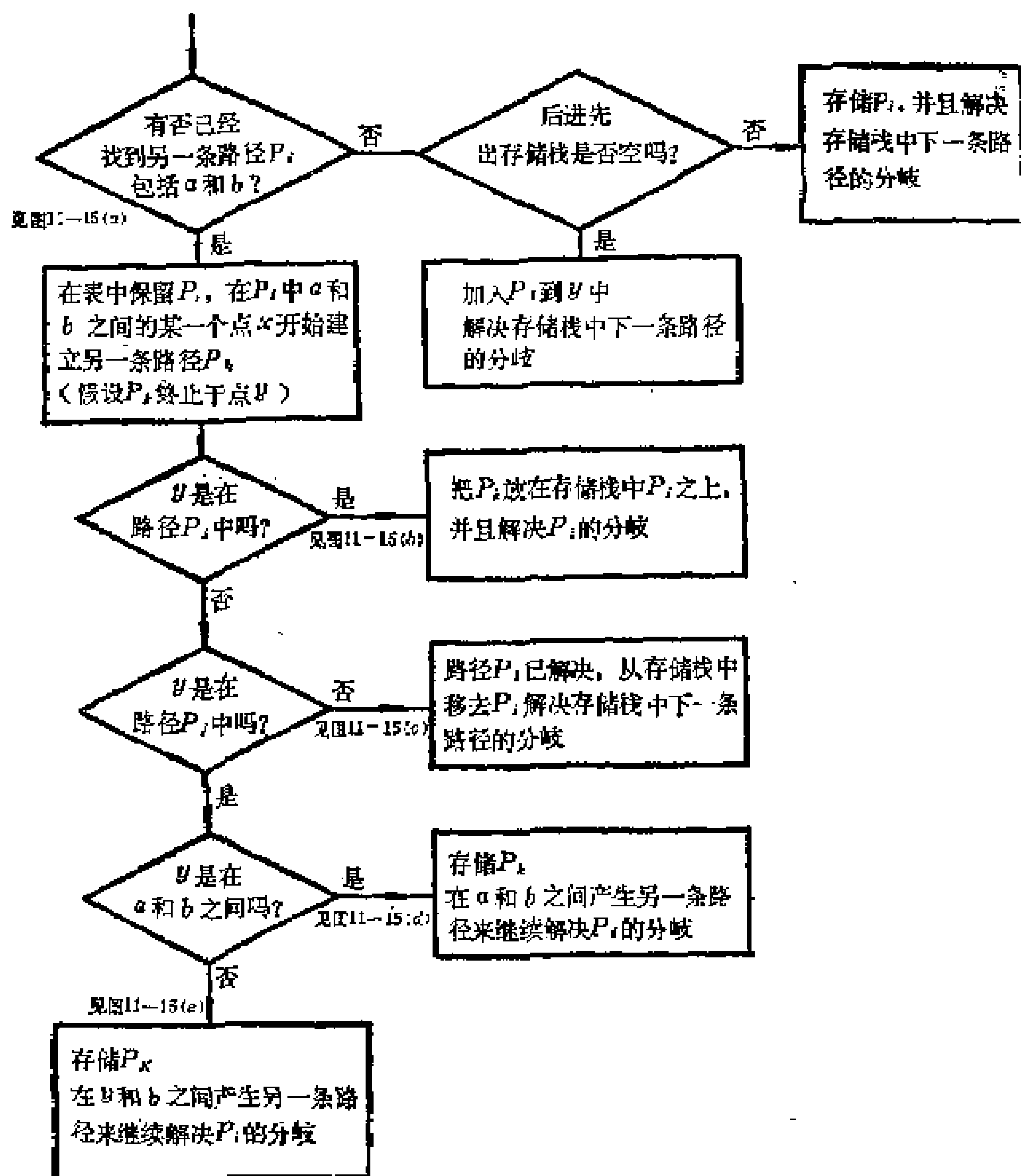


图 11—14

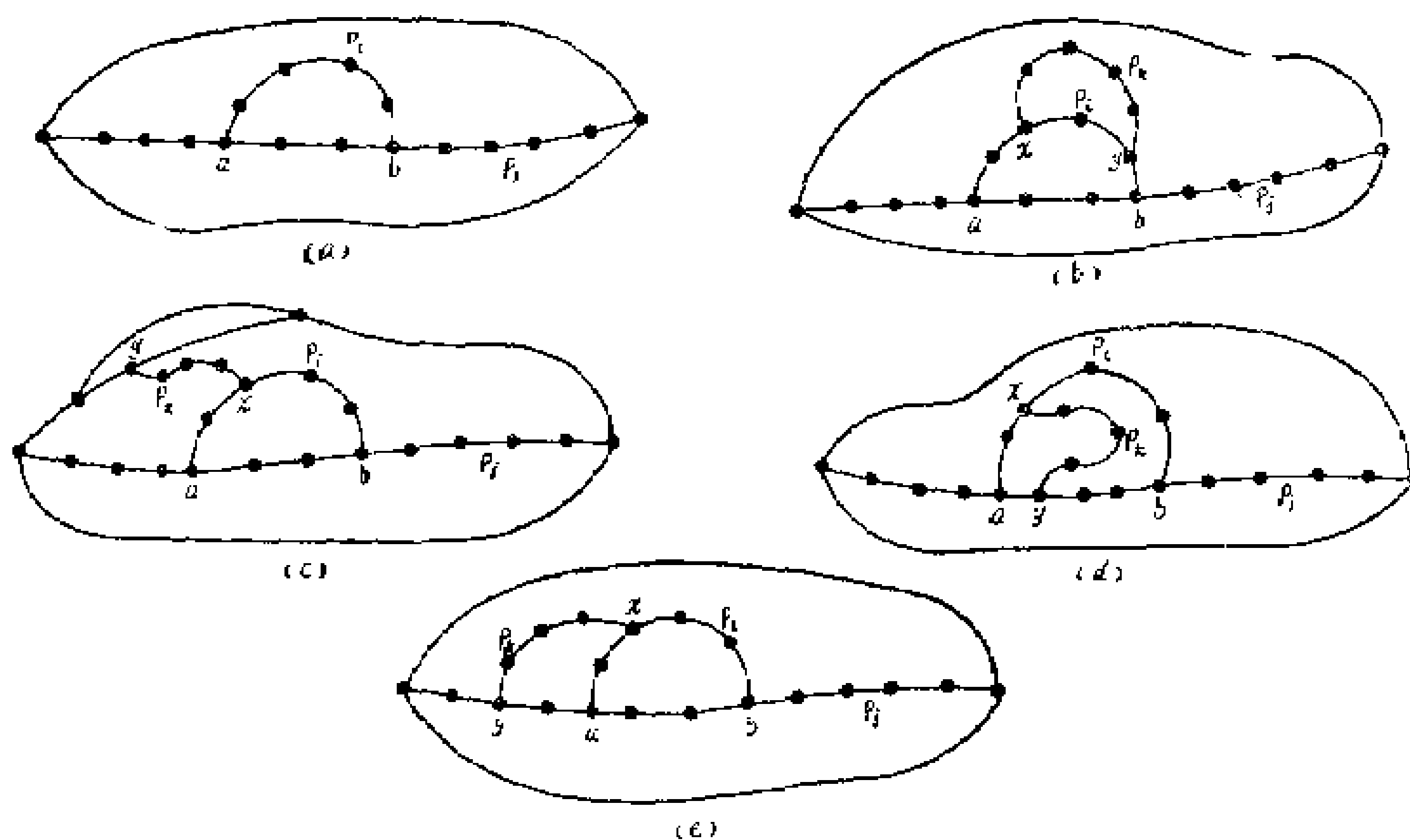


图 11—15

概括地讲，平面性算法包括下列程序：（1）找寻部分，（2）把部分划分为路径，

（3）决定把这些路径放到哪里去，和（4）建立一个平面表示法。深度第一搜索法用于（1）和（2）。由棕榈树和叶子所产生的路径有几个重要特性用于（3）和（4），除（4）外，每步所需要的时间是与边数n_e成正比的。建立平面表示法所需要的时间是与$n_v \lg n_v$成正比的。其中n_v是点数。因此，整个算法所需要的理论时间是与$n_v \lg n_v$成正比的。

第八节 其它图论算法

上述各节所讲的算法虽然是很重要的，并是基本的，但是它们只是现有文献中关于图论算法的几个实例而已。这里当然不可能都包括它们。其它常用的图论算法有：

（一）在一个给出的有向图中找寻所有片断即最大强连通子图。

（二）在一个给出的无向图中找寻一条汉密尔顿路径（如果有的话）。

（三）在一个有向图中找寻所有有向汉密尔顿回路。

（四）在一个给出图中找寻一个最大完全子图。

（五）在一个连通加权有向图中，如果一条边的权代表通过这条边的最大流量的话，找寻这图中由某一个规定点到另一个点的最大可能流量。

（六）在一个给出的连通加权图中找寻最小权的汉密尔顿回路。这就是第二章第十三节中所讲的巡回检查工作的旅程问题。到现在为止还没有一个满意的解法。

（七）在一个给出的有向图G中，找寻从G中删除一个最小边组可以破坏所有的有向回路。这就是第七章第十节中所讲的消除循环法，但是当图的点数n_v增多时，计算时间依n_v的指数律上升。

（八）在一个给出的有向图G中，找寻从G中删除一个最小点组（而非上一问题的最小边组）可以破坏所有的有向回路。正象上一问题一样，还没有有效的算法。

其它算法不胜枚举，而且有些算法的效率还是很差的。所以这方面还有很多工作可做。

一个算法最好能保证解决问题的执行时间能够是n_v或n_e的某些常数幂。换句话说，执行时间t（对于可能的最坏的图）可以表达为

$$t \leqslant \alpha n_v^p \quad 或 \quad t \leqslant \beta n_e^q \qquad (11-9)$$

p和q值越小越好。这样一个算法（它的执行时间是由n_v或n_e的多项式为界）叫作多项式为界算法。例如，连通度和部分算法是多项式为界的，其中$P=2$。生成树算法、基本回路算法、截割点和可分性算法、最短路径算法和平面性算法也都是多项式为界的。不过有向回路算法却不是了。下表是些重要的多项式为界的图论算法和时间的界限：

问题	操作时间的界限
连通度和部分	n_v或n_e
生成树	n_e
最短生成树	n_v^2
基本回路组	$n_v^r (2\leqslant r\leqslant 3)$
截割点和部分	n_v^2或n_e
桥接边	n_v^2或n_e
两点之间的最短路径	n_v^2
所有点对之间的最短路径	n_v^3
传递闭包	$n_v^2 (2<\alpha<3)$

强连通度和片段　　　　　　　　　　　　n_v^2或n_e

平面性　　　　　　　　　　　　　　　　n_e

注意：因为对于一个简单图来说，$n_e \leqslant n_v(n_v-1)/2 \leqslant n_v^2/2$，所以所谓以$n_e$或$n_v$为界是相互可以转换的。另一个应注意的是虽然不同算法同样以n_v的某次幂为界，但是对于一个相同图，因为各个算法有不同的乘法常数，计算机实际使用时间可能差别很大。

另一方面，有些图论算法的确不可能有一个多项式为界算法。例如，在一个给出图产生所有生成树的问题中，一个有n_v个点的简单标记树的生成树可能多至$n_v^{n_v-2}$。所以这个问题是找不到一个多项式为界算法的。同样地，对于一个给出图要产生所有完全子图，所有回路，所有有向回路，所有路径，所有截割组等等，也有同样问题。

另有第三类图论问题，它们一直到现在还没有发现有多项式为界算法，也无法证明这些问题是不会有多项式为界算法的。同构性检查或算法就是这样的问题。其他如汉密尔顿回路，有向汉密尔顿回路、最大完全子图，巡回检查的旅程问题。消除循环法中最小边组或点组等等都属于这类中的重要问题。解决这些问题（用现在可用的最好算法和可能最坏图）的计算时间随着点数n_v的增长而依指数律或阶乘律增长。这样效率差的算法当然对实际问题的用途是很有限的。启发式方法就成为解决这些问题的主要依靠了。

这里介绍一种拉丁乘法，可以从点数n_v少的有向图的邻接矩阵找出图的有向汉密尔顿路径。如果这个图没有汉密尔顿路径，那么可以找到n条简单的有向路径，包括所能到达的最长有向路径。这个方法也适用找无向图的汉密尔顿路径。现在把运算法说明如下：

第一步：用下法列出图的邻接矩阵M^1

（一）如由点i到点j之间有一条有向边，就是$x_{ij}>0$，$i\neq j$，则在矩阵的（i，j）处写出i，j。

（二）否则置零。

（三）矩阵的非零元素值称为串。

例如，以图11—16为例，它的邻接矩阵M^1为

$$M^1=\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \left(\begin{array}{ccccc} 0 & 1,2 & 1,3 & 0 & 0 \\ 0 & 0 & 2,3 & 0 & 0 \\ 0 & 0 & 0 & 3,4 & 3,5 \\ 0 & 0 & 0 & 0 & 4,5 \\ 5,1 & 5,2 & 0 & 5,4 & 0 \end{array}\right) \end{array} \qquad (11-9)$$

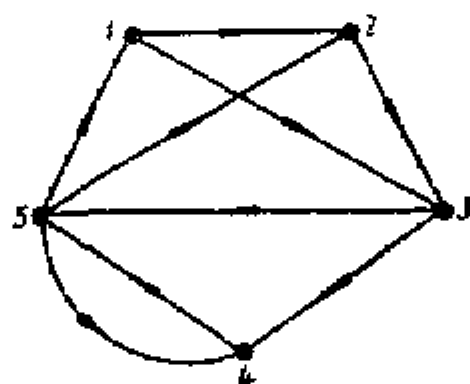


图　11—16

第二步：定义一个$n_v\times n_v$矩阵N^1。N^k是在M^k中删除每一个非零项的第一个点而得来的

第三步：$M^k \mathscr{L} N^1 = M^{k+1}$

式中符号$\mathscr{L}$代表拉丁乘法。这种乘法与普通矩阵乘法相似，规定如下：

（一）零乘任何项等于零。

（二）串乘法是将各串串联起来。例如

$$1,5,6\times 2,3,4=1,5,6,2,3,4,$$

（三）串加法是将加串写在被加数的下方。例如

$$1,5,6+1,3,4=\begin{array}{l}1,5,6\\1,3,4,\end{array}$$

(四) 任何一串中有多属一个同一点者等于零。第四步、矩阵M^k的元素就给出段数为k的简单路径。M^{n-1}给出汉密尔顿路径。

以图 2—36的有向图为例，应用拉丁乘法求这图的汉密尔顿路径。第一步的结果，即这个图的邻接矩阵 M^1如式（11—9）所示。第二步的矩阵 N^1如下，这里和此后为了简单起见，不再写出矩阵中的零元素 0。

$N^1=$

	1	2	3	4	5
1		2	3		
2			3		
3				4	5
4					5
5	1	2		4	

第三步中，

$M^2=M^1\mathscr{L}N^1=$

	1	2	3	4	5
1			1,2,3	1,3,4	1,3,5
2				2,3,4	2,3,5
3	3,5,1	3,5,2		3,5,4	3,4,5
4	4,5,1	4,5,2			
5		5,1,2	5,1,3 5,2,3		

在这一步，我们可以从$M^2\mathscr{L}N^1$得到M^3；或者，我们从M^2定义N^2，然后得 $M^4\mathscr{L}N^2$。如用后一方法，我们得

$N^2=$

	1	2	3	4	5
1			2,3	3,4	3,5
2				3,4	3,5
3	5,1	5,2		5,4	4,5
4	5,1	5,2			
5		1,2	1,3 2,3		

现在用$M^2\mathscr{L}N^2$来得到M^4，结果得到如下的矩阵

$M^4=$

	1	2	3	4	5
1		1,3,4,5,2		1,2,3,5,4	1,2,3,4,5
2	2,3,4,5,1				
3		3,4,5,1,2			
4			4,5,1,2,3		
5				5,1,2,3,4	

这个图的点数为 5，M^4就给出了有向汉密尔顿路径。所以我们得到图 2—36的 7 条有向汉密尔顿路径如下：

1,3,4,5,2　　1,2,3,5,4　　1,2,3,4,5　　2,3,4,5,1

3,4,5,1,2　　4,5,1,2,3　　5,1,2,3,4

在这7条有向路径中，后5条就是一个有向汉密尔顿回路1，2，3，4，5，1的中断而已。

第九节　专用计算机语言和程序设计

由于近年来对于图论计算的兴趣高涨，导致了专用处理线性图的若干种程序设计语言的发展。这样一种语言的主要目标在于使用户能用一种简洁自然语言来写成对线性图运算的公式，正象他与另一个图论学者通信一样。

这样一种语言，经研制和实现时，它的优越性是很明显而巨大的。它使图论程序的编写容易简练，使用户不必冗余不必要的细节而可专注于程序的实质方面。可是这种语言，也像其它专用语言一样有它的缺点，就是研制这样一种语言是费时，费力和费钱的，但是却只能用来编写图论方面的程序。

为了使一种图论计算机语言对很多用户解决图论中的不同问题有用，语言必须具备大量基本语句，如象"移去G中的点p"。"在点x和y之间加入一条边"，"在G中找寻由点u到点v间最短距离"等等。而且，图论语言还必须有所有现在已有的符号语言，如$FORTRAN$的计算方便条件等。这样才能使程序设计人员完成基本语句中未包括的各种作用。因为要从无到有研制一种全新语言并没有什么好处，所以现在所有的图论语言差不多都是几种著名的程序设计语言的发展。例如，图论语言GTP是$FORTRAN$的发展；图论算法软件组$GASP$是$PL/1$的发展；图论发展算法语言GEA是$ALGOL$的发展等等。这里不预备列举现在可以采用的图论语言，有兴趣的读者可查阅这方面的文献。

现在来谈谈线性图在程序设计中的用处。从计算机的程序设计的早期，一种程序的分析就是一个重要问题。这样一种分析的目的在于估计操作时间、一个程序的存储量要求、将一个大程序划分为若干个子程序、检查程序中某些结构错误、为一程序提供文件、或者只是为了理解别人编写的程序。为了所有这些目的，用一个有向图来代表一个程序是一种很方便的形式。图中每一点代表一个程序块，就是一序列的计算机指令。只要每当序列中的任何一个指令执行了，所有指令就都执行了。每一程序块有一个进入点和一个出口点，前者是序列中的第一个指令，后者是序列中的最后一个指令。每一条边(v_i, v_j)代表从程序块v_i的最后一个指令到程序块v_j的第一个指令间的可能控制转移。这样一个有向图称为一个程序有向图，也可以看作是一个框图的抽象化，其中逻辑框缩小成点，箭头变成有向边。例如图11—17是框图11—9的程序有向图〔现在暂时不管虚线(v_{14}, v_1)〕。

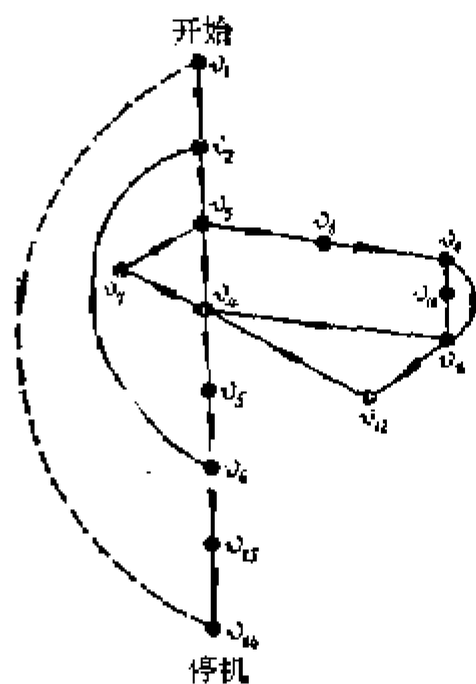


图 11—17

注意，因为一个程序块可能分解为几块，所以一个程序可能有几个程序有向图。也要看到有些程序在程序有向图中是表示不出来的，如像在执行过程中改变它们的控制和处理指令，在程序有向图中也忽略了说明语句，象数据安排的格式和维数语句。

代表任何一个正确的计算机程序的有向图有下列一些明显而重要特性：

（一）一个程序有向图一定是连通的。

（二）它恰好只有一个点的入次数为零，这个点对应着程序的开始。如程序有几个开始点可以再增加一个开始点，而用有向边连到这些点上。

（三）它恰好只有一个点的出次数为零，这个点对应着程序结束。我们叫这个点为停机点。如果在程序中有几个停机点，可照上一项处理。

（四）程序有向图中每一个点都可以从开始点到达的。

（五）停机点可以从程序有向图中任何一个点到达的。

程序设计错误的检查，编译程序工作的一个必不可少的部分是检查和报告源程序中某些结构错误，这些错误中最常见的可以在程序有向图中从开始点到停机点的有向路径中跟踪检查。路径跟踪可以检查出没有放进去的程序块或者没有出口导致停机点的程序块，或者脱节部分，如像无从调用的一个子程序。

程序操作时间的估计：给出一个计算机程序和每一程序块的执行时间后，我们常常需要估计一下这个程序的操作时间。这种情况可由一个程序有向图来代表，而图中的每一个点v_i结合着一个权t_i，这个t_i就是对应程序块的执行时间。如果我们可以估计每一个点通过的次数（就是程序块执行次数），整个程序的操作时间就可以确定了。当然，这是对某一特殊计算机而言的。

在程序有向图中，设每一条有向边e_i指定一个正整数f_i，这个f_i是穿行边e_i的次数。除了开始点和停机点外，进入一个点的次数一定等于离去的次数。考虑到这两个特殊点，我们可以从停机点v_{n_v}引入一条有向边到开始点v_1，如图11—17中虚线所示，它的权为1。现在这个修改了的有向图的那些边权可以在每一点都满足克希荷夫电流定律，其中f_i可以看作是通过边i的流量e_i。

给出一个连通有向图G，它有点v_1，v_2，…，v_{n_v}和n_e条边，边权分别为f_1，f_2，…，f_{n_e}。现在要问G对应着某个有v_1为开始点和v_{n_v}为停机点的程序有向图的必要和充分条件是些什么？有些必要条件是明显的，每个f_i一定是个正整数，v_1的入次数$d^-(v_1)=0$；v_{n_v}的出次数$d^+(v_{n_v})=0$；除了在v_1和v_{n_v}外，其它点的那些f_i一定满足克希荷夫电流定律。另外，从点v_1出来的那些边权之和应该等于进入v_{n_v}的那些边权之和，两者都等于1。那么这些条件充分吗？从下面的构图中，我们的答案是"充分"的。

从给出的加权有向图G，让我们如下构成一个无权有向图H：用f_i条平行边代替每条边e_i，其中f_i是有向图G中边e_i的权。很清楚，当且仅当G中每一个点能够满足克希荷夫电流定律时，有向图H是平衡的。就是说，H中每一个点v的$d^+(v)=d^-(v)$。由定理7—8知，当且仅当一个有向图是一个有向欧拉图时，这个有向图是平衡的，就是说，在H中有一条从v_1到v_{n_v}的有向欧拉行走。当且仅当在G中从v_1到v_{n_v}有一个有向边序列，其中每条边e_k正好出现f_k次，而边(v_{n_v},v_1)在这个有向序列中没有出现时，才有可能有这样的一个有向欧拉行走。这最后一句话等于是说，G是一个程序有向图，其中v_1是开始点，v_{n_v}是停机点。

程序分段：有时候，人们可能碰到一个程序，它是大到使现用计算机的暂时存储器无法整个容纳它。在这种情况时，在执行这个程序之前，必须把程序分段。这时这个程序只能存储在大容量外存储器（如磁鼓，磁盘或磁带）中，而在执行时一次从外存储器取一个程序段。每个程序段的大小要选择适当，即要小到足以容纳在暂时存储器中，也要大到在快的暂时存储器与慢的外存储器之间不必太多的转移。因此我们有一个找寻程序有向图最优划分成有向子图的问题。要使一些点的权之和（这里点v_i的权S_i是程序块i所需要的存储空间）不

要超过某一规定值。在多路存取分时系统中也有同样问题。就是每一户有一固定时间的使用分段，程序段必须选择得适宜，一方面不要长到执行时超过规定时间，另一方面也不要短到在内外存储器间需要低效率的转移。

如果程序有向图是非循环的（就是程序中没有回线，这对于重大的计算机程序是少有的），划分问题就很容易解决。我们可以把那些点按着图形顺序由第一个点开始加以分类，我们把序列划分成最大可能的子序列，使得每一个子序列的总点权不超过一个规定值。

程序分段的困难之处在于程序有向图有些有向回路（就是程序中的回线）对一个有向回路的截割意味着两个程序段之间的互换，也就是快慢存储器之间的互换。因此人们还是喜欢避免程序分段时使一个有向回路的一些点属于一个以上的分段。完成这个工作的最简单方法是在程序有向图中标记所有片段（即最大强连通有向子图），然后把每个片段中所有点的权加起来计算它的存储器要求。如果这些存储器要求中的最大者没有超过暂时存储器的容量，那么问题解决了。这样我们不需要截割任何有向回路，而小的片段总可以合并起来得到合理大小的程序段。

可是在实际上会发现最大的片段常常对于暂时存储器还是太大的。例如，在图11—17中，差不多整个程序有向图组成一个片段。在这种情况时，截割有向回路是避免不了的。曾有人建议过最简单的途径是找出一个分离最少数有向回路的划分方法。可这是一个困难问题。这是因为在一个程序有向图中列举所有片段，并且把每个片段作为一个可能的程序段是一项巨大的任务。我们已经在算法中看到，即使在一个有向图中产生所有的有向回路也是极费时间的。

一个程序的分段至少是一个很难的问题。对于这个重要和有趣的图论问题，到现在为止还没有找到一个有效的解决方法。

习　　题

11—1　当输入图 7—4（a）的数据时，试比较用下列 4 种方法时分别各法存储量的大小：

（1）邻接矩阵，（2）关联矩阵，（3）列举各边法，（4）两个线性数组法。

11—2　当一个图用关联矩阵 A 或邻接矩阵 X 表示时，试设计程序：

（1）由关联矩阵转变成邻接矩阵；

（2）由邻接矩阵转变成关联矩阵；

（3）由关联矩阵转变成基本回路矩阵。

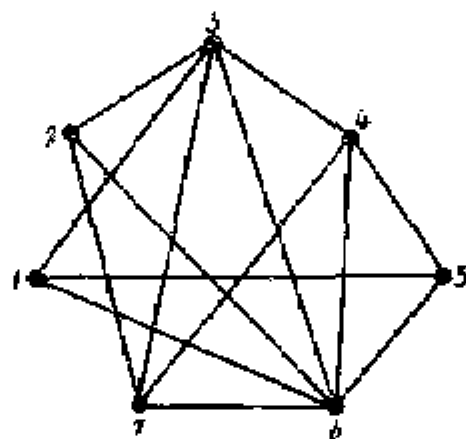


图 11—18

11—3　试由电子计算机求图 3—18中以点 v_1 为根的最短生成树。

11—4　求图 3—18的所有生成树。

11—5　试设计一个程序，求从一个规定点到另一规定点之间的最短路径所经历的各点。通过计算机求图11—8 从点 A 到点 G 之间最短路径所经历的各点。

11—6　试设计一个程序，求一个图中每对点之间最短路径所经历的各点。通过计算机对图11—8 求各对点之间的最短路径所经历的各点。

11—7　应用拉丁乘法求图11—18中无向图的汉密尔顿路径。

第十二章　边权交通网络

第一节　无向边权交通网络的流量

由一站传输信息到另一站的介质（如电话线路、铁路、公路、煤气管路、电力线路等）所组成的网络叫做交通网络。它可以用一个线性图来代表，其中每一个点表示一个站，每一条线表示两站间传递信息的介质。网络所能传递的最大信息量仅受介质而非站所限制的网络叫做边权交通网络。网络所能传递的最大信息量仅受站（或点）所限制的网络叫做点权交通网络。这一章我们只讨论前一种情况即边权交通网络，而把后一种情况留到下一章中讨论。

因为在边权交通网络中，只是那些边限制着信息的传输，只有边有容量或权，它表示着边所能传输的最大信息量。这些权叫做边负载量。当所有边无定向时，这个网络叫做无向边权交通网络，而当所有边有定向时就叫做有向边权交通网络。例如，图12—1所示的线性图是一个无向边权交通网络，其中各边上括弧中的权是边的负载量。对负载量而言，我们可以不说是由点 i 到点 j 传输的信息量而说是由 i 到 j 所分配或指定的流量。

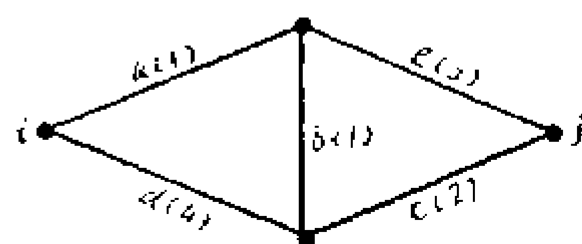


图 12—1

设 P_r 是在一个无向图中点 i 和 j 之间的一条路径。给予 P_r 中每一条边一个从 i 到 j 的定向，我们就得到一条从 i 到 j 的对应有向路径。

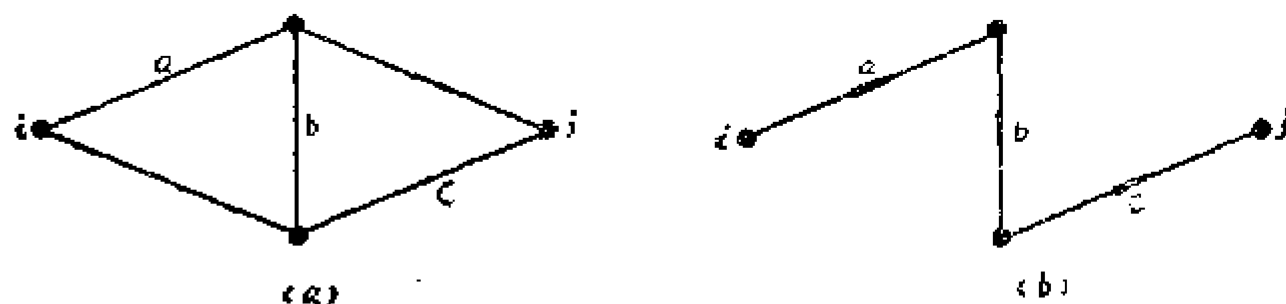


图 12—2

例如在图12—2（a）的无向图中一条路径 $P=$（a、b、c），它的对应有向路径就如图12—2（b）所示。

符号 $\psi_{r_{ij}}$ 表示由点 i 到点 j 的一个流量，符号的下标就指明起点和终点。所谓对一条路径分配一个流量 $\psi_{r_{ij}}$ 的意思就是说在这条路径中的各边传输着 $\psi_{r_{ij}}$。

设 P_r 是一个无向图中点 i 和 j 之间的一条路径。对 P_r 分配一个流量 $\psi_{r_{ij}}$，意思就是说在 P_r 的每条边中分配 $\psi_{r_{ij}}$ 它的定向是与 P_r 的从 i 到 j 的对应有向路径一致的。

例如，对路径 $P=$（a、b、c）分配 $\psi_{r_{ij}}=1$，我们给 P 中每一条边一个适当定向的一单位流量，如图12—3（a）所示。为了简单起见，一般用一条虚线来表明 $\psi_{r_{ij}}$ 如图12—3（b）所示。

当我们说对一个无向边权交通网络G分配一个流量 $\psi_{r_{ij}}$ 时，一般我们指

$$\psi_{ij}=\sum_{r=1}^{n}\psi_{r_{ij}} \tag{12—1}$$

其中流量ψ_{ij}的每一部分$\psi_{r_{ij}}$是分配给G中点i和j之间的一条路径P_r，$r=1，2，\cdots，n$，和$\sum_{r=1}^{n}\psi_{r_{ij}}$是指$\psi_{r_{ij}}$的总和。

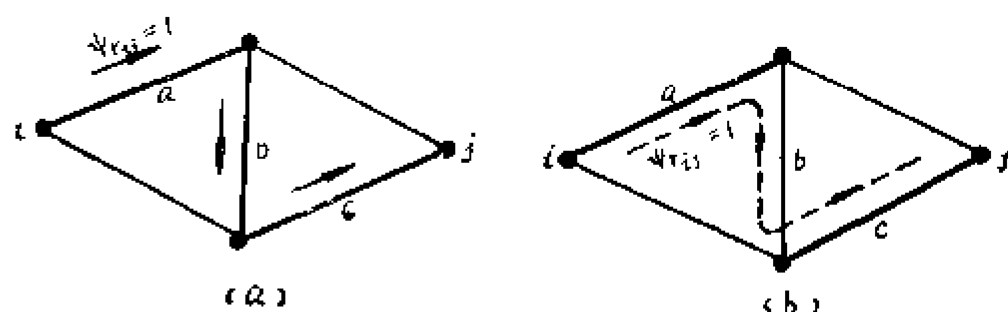


图 12—3

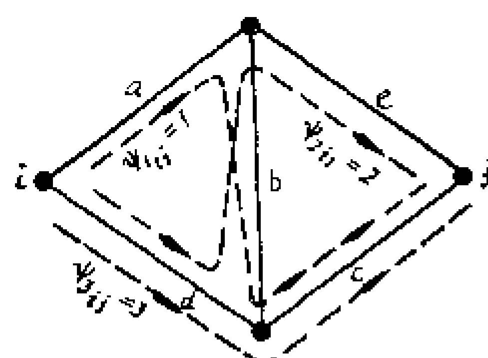


图 12—4

例如图12—4中的无向边权交通网络G由点i到j有3个流量：$\psi_{1_{ij}}=1$，$\psi_{2_{ij}}=2$；$\psi_{3_{ij}}=3$分别分配给路径P_1，P_2和P_3，我们就说分配给G的总流量$\psi_{ij}=\sum_{r=1}^{3}\psi_{r_{ij}}=6$。

必须注意到，从边负载量的定义可知，分配给路径P_r（$r=1，2\cdots，n$）的流量$\psi_{r_{ij}}$不应该超过边的负载量。例如在图12—4的边权交通网络中，边b必须负载$\psi_{1_{ij}}+\psi_{2_{ij}}=3$。如果边$b$的边负载量小于3，那么图12—4中所表示的分配量就不合适，就不能达到$\psi_{ij}=6$。换句话说，对每一条路径P_r给予流量$\psi_{r_{ij}}$是有某些限制的。

假使分配给一些路径$P_{s_{ij}}$从i到j所有s的流量是些$\psi_{s_{ij}}$，（$s=1，2\cdots，h$），那么对于任何一条边负载量为c的边e必须满足下列方程

$$c\geqslant\psi(e) \tag{12—2}$$

$$\psi(e)=\sum_{s=1}^{h}\delta_s(e)\psi_{s_{ij}} \tag{12—3}$$

$$\left.\begin{array}{l}\delta_s(e)=1，\text{如果边}c\text{是在路径}P_s\text{中}\\ \delta_s(e)=0，\text{如果不在}P_s\text{中的话}\end{array}\right\} \tag{12—4}$$

式（12—2）中的符号$\psi(e)$表示对诸路径$P_{s_{ij}}$（$S=1，2\cdots，h$）所分配的流量$\psi_{s_{ij}}$中分配给边e的总流量。

假使分配给诸路径$P_{s_{ij}}$（$s=1，2\cdots，h$）的诸流量为$\psi_{s_{ij}}$，我们知道边权交通网络G中每一条边都必须满足式（12—2）。为了对每一条径$P_{t_{ij}}$再分配给一个额外流量$\psi_{t_{ij}}$，在径路$P_{t_{ij}}$中每一条边e_u必须满足

$$\psi(e_u)+\psi_{t_{ij}}\leqslant c_u \tag{12—5}$$

式中$\psi(e_u)$是式（12—3）所给出的边e_u中的总流量，c_u是边e_u的负载量。式（12—5）可以改写成

$$\psi_{t_{ij}}\leqslant c_u-\psi(e_u) \tag{12—6}$$

这是$P_{t_{ij}}$中每一条边必须满足的条件，换句话说

$$\psi_{t_{ij}}\leqslant\min\{c_u-\psi(e_u)\,|\,e_u\in P_{t_{ij}}\} \tag{12—7}$$

式中右边项给出对路径$P_{t_{ij}}$分配一个额外流量的上限。

例12—1　假使给予图12—1的边权交通网络的流量ψ_{ij}如图12—5所示。由图各边的边负载量可见，这里只有一条路径$P_{ij}=(e_4, e_5)$还可以分配给以从i到j的非零流量。因为$\min\{4-\psi(e_4); 2-\psi(e_5)\}$为1，所以再分配给的额外流量$\psi_{2ij}$不能大于1。

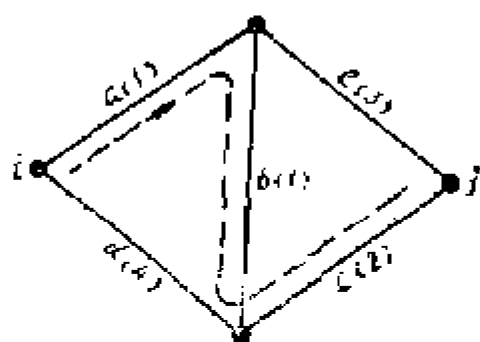


图　12—5

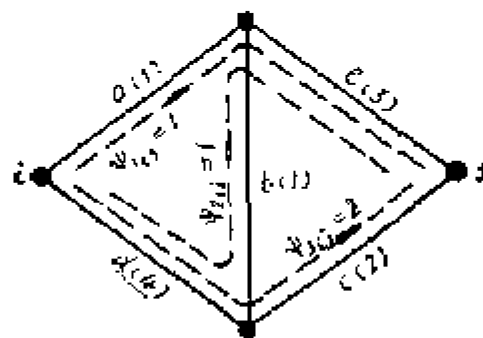


图　12—6

所谓"可以分配的流量ψ_{ij}"是指有一组路径P_{rij}（$r=1, 2\cdots, h$），其中每条径路是在点i和j之间，并有一组流量ψ_{rij}，其中$\psi_{ij}=\sum_{r=1}^{h}\psi_{rij}$，这些流量$\psi_{rij}$可以按$r=1, 2\cdots, h$）顺序分配给这些路径$P_{rij}$。

在例12—1中，我们可以知道总流量$\psi_{ij}=2$可以分配给图12—1所示的交通网络。可是如果我们不象图12—5那样分配$\psi_{1ij}=1$的话，我们可以象图12—6那样分配总流量$\psi_{ij}=\psi_{1ij}+\psi_{2ij}+\psi_{3ij}=4$。这就表明所分配的总流量是根据流量如何分配而定的。另一方面，如果一个边权交通网络包括有限边数，并且每个边负载量是有限的话，我们知道只能分配一个有限流量。因此从点i到点j所能分配的流量一定有一个最大流量。这个最大流量叫做从i到j的极限负载量，符号为t_{ij}；它的定义为：

一个边权交通网络G的极限负载t_{ij}是所能分配给G中从i到j的最大流量。

假使一个流量ψ_{ij}可以分配给一个边权交通网络G，并且假定ψ_{ij}是最大流量，就是ψ_{ij}等于从i到j的极限流量t_{ij}时，那么只要分配给路径P_{rij}（$r=1, 2\cdots, h$）的$\psi'_{rij}=\alpha\psi_{rij}$可以满足式(12—2)，式中

$$\alpha=\psi'_{ij}/\psi_{ij} \tag{12—8}$$

$$\psi_{ij}=\sum_{r=1}^{h}\psi_{rij} \tag{12—9}$$

那么任何流量$\psi'_{ij}\leqslant\psi_{ij}$可以分配给这个网络$G$，所以知道一个边权交通网络的极限负载量$t_{ij}$是重要的。为了讨论极限负载量，我们需要下列定义。

如果通过一条边的流量ψ_{ij}等于这条边的负载量时，在分配这个流量给这条边时，这条边就叫做饱和边。

例如在图12—5中那样的流量分配中，边e_1和e_2都是饱和边，在图12—6中，边e_1，e_2和e_5都是饱和边。

在流量ψ_{ij}的分配下，如果一个截割组S包括些饱和边，那么S叫做饱和截割组。

例如在图12—6中，边e_1，e_2和e_5都是饱和边，所以$S=(e_1, e_2, e_5)$在所分配的流量ψ_{ij}下是一个饱和截割组。我们选择一个饱和截割组的定向等于截割组中的某一条边的流量方向。对一条边所分配的几个流量可能不是同一方向的。

在图12—7中有一个边权交通网络，在边e_2中有两个方向相反的流量。在e_2中取其中的

一个流量，譬如ψ_i作截割组S的定向。我们可以说在边e_1中的流向是与S的定向不相符的。同样，我们也可以看出边e_5中的流向也是与S的定向不同的。

如果一条边是饱和的，并且在这条边中所有非零流量都是同一方向（定向）的，那么这条边叫做基础饱和边。例如在图12—7中边e_1和e_5都是基础饱和边，而边e_2却不是。

当在一个饱和截割组S中所有各边都是基础饱和边，并且截割组的定向是与截割组中每条边中的非零流向一致时，这个S叫做基础饱和截割组。很明显，图12—7中S以及图12—8中的饱和截割组S都不是基础饱和截割组，而图12—6中的饱和截割组（e_1，e_2，e_5）才是一个基础饱和截割组。

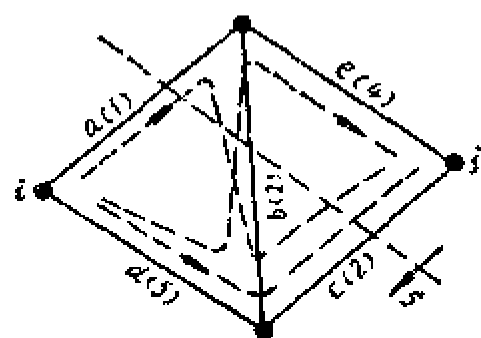


图 12—7

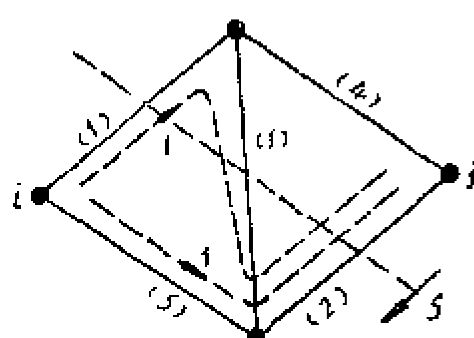


图 12—8

在图12—8中有一个边权交通网络，其中S是一个饱和截割组，但不是一个基础饱和截割组。这就很容易看出，从i到j有一条路径P_{ij}，对它分配了一个非零流量ψ_{rij}，并且这条路径至少含有截割组S中的三条边，如图12—9所示。另一方面，如果S是一个基础饱和截割组，那么从i到j的每一条路径P_{ij}，对它分配一个非零流量ψ_{rij}，一定正好含有S中的一条边，如图12—10所示。

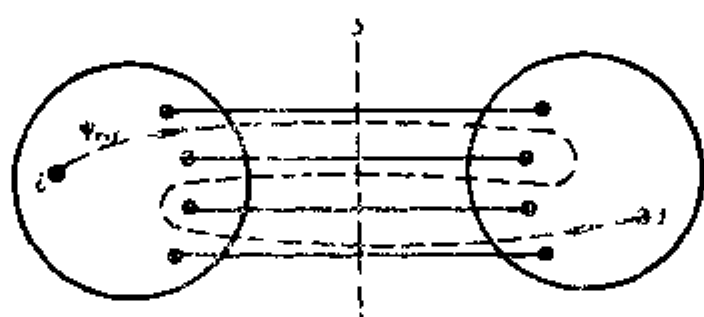


图 12—9

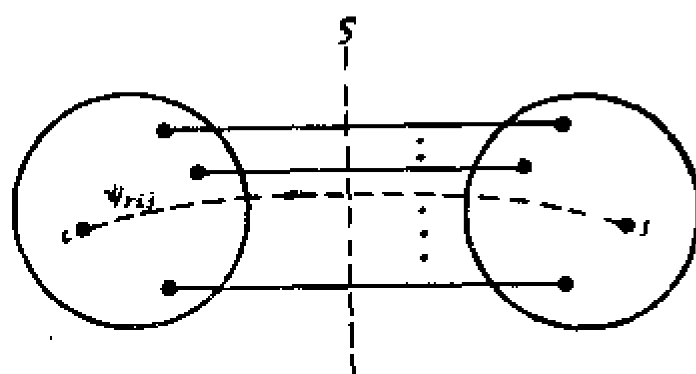


图 12—10

第二节 最大流量

在研究最大流量与基础饱和截割组的关系以前，让我们先研究有了一个饱和截割组对分配额外流量的影响。

辅助定理12—1：对一个边权交通网络分配一个流量Ψ_{ij}后，如果没有饱和截割组，那么从i到j可以分配额外流量。

证：在一个边权交通网络中，因为没有饱和截割组时，如果我们把每条边e_r的边负载量c_r改变为$c'_r = c_r - \Psi(e_r)$，式中$\Psi(e_r)$是分配给边e_r的总流量，并且删除新的边负载量c'_r为零的那条边，那么在结果网络中从i到j一定至少有一条路径P_{ij}，否则就一定有一个饱和截割组了。因为在路径P_{ij}中每条边的边负载量c'_r不是零，我们就可以对路径分配一个非零流量ψ'_{ij}，其中

$$\Psi'_{ij} \leqslant \min\{c'_r \mid e_r \in P_{ij}\} \quad (12—10)$$

根据辅助定理12—1，如果我们对一个边权交通网络G不能再分配些流量ψ'_{ij}，我们知道其中至少有一个分开点i和j的饱和截割组。

我们在第四章第五节已经知道如何得到一个图的所有截割组。但是如何得到分开一个图中某两个规定点的所有截割组呢？为了这个目的，我们先介绍两个符号$S(i,\ j)$和$S(ij,\ \ )$。

如果截割组$S=E(V_1\times\overline{V}_1)$，其中$i\in V_1$，$j\in\overline{V}_1$，那么这个截割组$S$是分开点$i$和$j$的，用符号$S(i,\ j)$来表示。如果$S=E(V_1\times\overline{V}_1)$其中点$i$和点$j$两者都在$V_1$（或$\overline{V}_1$）之中，那么这个截割组$S$不能分开$i$和$j$的，用符号$S(ij,\ \ )$来表示。注意这里的$S$可以是一个截割组或者是若干截割组的一个边不共用并集。

所谓截割组集体$\{S\}$是包括一个图的所有可能的截割组的，其中当然也包括分开某两个规定点的那些截割组的。$\{S\}$是可以把一个图的n_v-k生成元通过环和运算得到的。为了得到那些分开i和j的截割组，应该用什么类型的生成元通过环和运算来得到呢？事实证明$S(i,\ j)\oplus S_2(i,\ j)$和$S_3(ij,\ \ )\oplus S_4(ij,\ \ )$的结果不能分开点$i$和$j$，只有$S_5(i,\ j)\oplus S_6(ij,\ \ )$才可以分开点$i$和$j$。

假设一个图G中的每一个点表示一个岛，每条边代表连接两个岛的桥，现在的问题是找寻破坏最少数桥而能使岛i和j之间的路径都不通。假使我们把一个无权线性图中的一个截割组S所包括的边数称为截割组S的值，用符号$\cup(S)$来表示。再假使$\{S^\circ(i,j)\}$是分开点i和j的所有可能截割组集体，$\{S(i,j)\}$是分开点i和j的所有可能截割组以及截割组的边不共用并集的集体，那么为了分开i和j所需要破坏的最少桥数N_b为

$$N_b=\min\{\cup(S)\mid S\in\{S^\circ(i,j)\}\} \quad (12—11)$$

对于分开点i和j的截割组的边不共用并集S，（并集中的截割组不是都能分开i和j的）其中有一个确实分开i和j的截割组S'，它是S的一个真子组。因此

$$\begin{aligned}&\min\{\cup(S)\mid S\in\{S^\circ(i,j)\}\}\\&=\min\{\cup(S)\mid S\in\{S(i,j)\}\}\end{aligned} \quad (12—12)$$

因此我们只要产生$\{S(i,j)\}$中的所有组而不必问其中那些组是真正分开i和j的。例如在图12—11中的线性图G中，根据定理4—9，我们只需要n_v-1个线性无关关联组通过环和运算就足以求得这个图的所有截割组以及截割组的边不共用并集。这里$n_v=5$，所以我们可以选择下列4个线性无关关联组

$S(i)=(1,2,3,4)$

$S(j)=(5,6,7,8)$

$S(v_1)=(3,5,6)$

$S(v_2)=(2,7,9)$

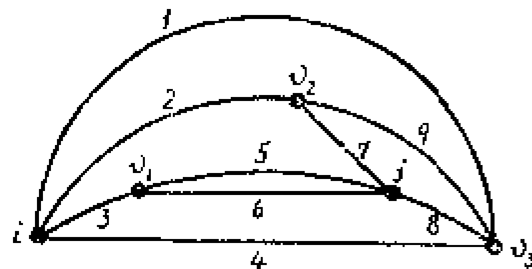


图 12—11

因为在$\{S(i,j)\}$中的任何一个组一定含有$S(i)$或$S(j)$，但不会含有两者。注意在上面的4个线性无关联组中，只有$S(i)$和$S(j)$是分开点i和j的。因此为了得到$\{S(i,j)\}$，我们可以取$S(i)$或$S(j)$与其余关联组的所有可能组合，如

$S(i)=(1,2,3,4)$

$S(i)\oplus S(v_1)=(1,2,4,5,6)$

$S(i)\oplus S(v_2)=(1,3,4,7,9)$

$$S(i)\oplus S(v_1)\oplus S(v_2)=(1,4,5,6,7,9)$$
$$S(j)=(5,6,7\ 8)$$
$$S(j)\oplus S(v_1)=(3,7,8)$$
$$S(j)\oplus S(v_2)=(2,5,6,8,9)$$
$$S(j)\oplus S(v_1)\oplus S(v_2)=(2,3,8,9)$$

这些就是分开点 i 和 j 的所有可能截割组。根据式（12—11）我们得到 $N_B=3$；就是应该破坏的桥是3，7和8。

下面是关于最大流量的一条定理。

定理12—1：分配给一个有限边数的边权交通网络 G 一个流量 ψ_{ij}，如果其中有一个饱和截割组而非基础饱和截割组时，那么

$$\psi_{ij}<t_{ij} \tag{12—13}$$

注意 t_{ij} 是从 i 到 j 的极限负载量，它等于从 i 到 j 的最大流量。

从定理12—1我们可以得到下一定理。

定理12—2：当且仅当一个流量 ψ_{ij} 的分配至少产生分开 i 和 j 的一个基础饱和截割组时，

$$\psi_{ij}=t_{ij} \tag{12—14}$$

证：当考虑图12—10中任何一个分开 i 和 j 的截割组时，我们可以知道流量 ψ_{ij} 一定通过截割组中的那些边的。因此从点 i 到 j 所能输送的最大量是不能超过截割组中各边的边负载量之和的，因为这对于分开 i 和 j 的任何截割组是正确的，所以对于分开 i 和 j 的一个基础饱和截割组也是正确的。而且通过一个基础饱和截割组的总流量等于截割组中各边的边负载量之和。因此式（12—4）是正确的。这式连同定理12—1就证明了这个定理。▮

上面已经讲过在无权线性图中截割组 S 的值 $\cup(S)$ 表示截割组 S 所包括的边数。在边权交通网络呢？所谓截割组 S 的值 $\cup(S)$ 是 S 中所有边的边负载量之和。

考虑分开点 i 和 j 的所有截割组的集体 $\{S_q(i,j)\}$，注意在 $\{S_q(i,j)\}$ 中任何一个截割组 S，它的 $\cup(S)$ 不能小于最大流量 ψ_{ij}。如果分配一个流量 ψ_{ij} 产生一个基础饱和截割组 S'（它分开着 i 和 j），S' 是在 $\{S_q(i,j)\}$ 之中。在一个基础饱和截割组 S' 中分配给各边的流量之和一定等于 ψ_{ij}。因此我们可以断定这符合定理12—2的结果。

现在把边权交通网络，也就是运输网络理论中一个最重要的定理叫做最大流量最小截割定理介绍如下。

定理12—3：对于一个边权交通网络，由点 i 到 j 的最大流量。即极限负载量等于网络中所有截割组中的最小负载量，即

$$t_{ij}=\min\{\cup(S)\mid S\in\{S_q(i,j)\}\} \tag{12—15}$$

证：在一个连接图 G 中由点 i 到 j 之间的任何一个截割组 S，当 G 中删除 S 后所余留下来的子图（$G\ominus S$）中，点 i 和 j 之间是没有路径的，即在 G 中点 i 和 j 之间的每条路径一定至少含有 S 中的一条边。因此由 i 到 j 或由 j 到 i 的每一流量一定经过 S 中的一条或几条边。所以在这两点之间的最大流量不能超过 S 的极限负载量。既然关于点 i 和 j 之间的所有截割组都是这样，所以由点 i 到 j 的最大流量不超过所有截割中的最小负载量。▮

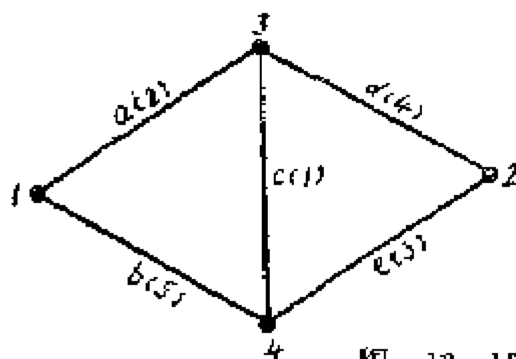


图 12—12

例12—2　在图12—12中有一个边权交通网络。分开点1和2的截割组集体是

$$\{(a,b),(a,c,e),(b,c,d),(d,e)\}$$

所以 $t_{12}=\min\{\bigcup(a,b),\bigcup(a,c,e),\bigcup(b,c,d),\bigcup(d,e)\}=\min\{7,6,10,7\}=6$

第三节　通信网络的图表示法

在通信网络中，有选择性地由一点接通到另一点，因此这个网络可以用一个线性图来表示，其中点代表些能够接通的端站或设备，而边是些联接线，链路或信道。所谓端站当然是可以接通的，不过所有接通差不多都要通过些中间点的。所以这个图虽然是一个连通图，但是一般地讲是大而复杂的。

在通信网络中，包括计算机网络在内，也象其它交通网络一样，重要的研究问题是联接线能否通过业务量的问题。所以很大一部分问题是属于边权交通网络理论的。因此图论方法可以用以研究通信网络中的选路方法和积聚的可能性。

在最简单的通信网络中，任何两个指定点之间只有一条唯一的路径。但是在现代的电话系统中是有很多变通路径的。利用图的路径，我们可以计算这些变通路径数，由此而研究系统的特性，以避免阻塞和决定选路算法。

一个指定网络的积聚问题基本上是一个组合学问题，需要计算网络中的阻塞状态数和非阻塞状态数。虽然组合学方法已经广泛地应用在现代话务理论，但是一般只限于减化了的情况，如忽略了相继机键级之间占有的互相依赖关系，所以是不精确的。这里只预备介绍些图论在通信网络中应用的基本知识，至于深入的研究可在一些专门文献中去寻找。

通信网络靠所选的通信联接线把一对用户连成一个通路。譬如说，用户A要求接通用户F。要满足这个要求，可能要通过几个交换局或电话局B、C、D、E转接，如图12—13所示，其中A，G、H，I，L，M，N等代表用户或端机，B，C，D，E、J、K等代表交换局。用图论的表示方法，我们可以用点代表用户、端机或交换局，用边代表通信联接线或通信链路，如图12—14所示。当用图论的抽象表示法时，我们就不管所用的技术手段和地

$$X=\begin{array}{c}\\ A\\ B\\ C\\ D\\ E\\ F\\ G\\ H\\ I\\ J\\ K\\ L\\ M\\ N\end{array}\begin{array}{c}\begin{array}{cccccccccccccc}A&B&C&D&E&F&G&H&I&J&K&L&M&N\end{array}\\ \left(\begin{array}{cccccccccccccc}
0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&1&0&0&0&0&0&0&0&0&0&0&1\\
0&1&0&1&0&0&0&0&0&1&1&0&0&0\\
0&0&1&0&1&0&0&0&0&1&0&0&0&0\\
0&0&0&1&0&1&1&1&1&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&1&1&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&1&1&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0
\end{array}\right)\end{array}$$

理距离，只是关心点和边的关联关系，换句话说，我们只关心哪些点相连接着。用图12—13的实际网络与图12—14的线性图之间有着明显的同构性。在图12—14中，这种关系可以用下列的邻接矩阵X表示出来。

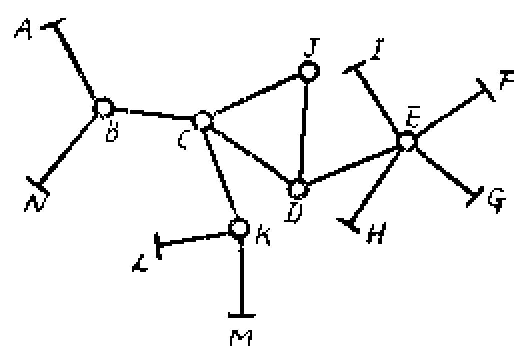


图 12—13

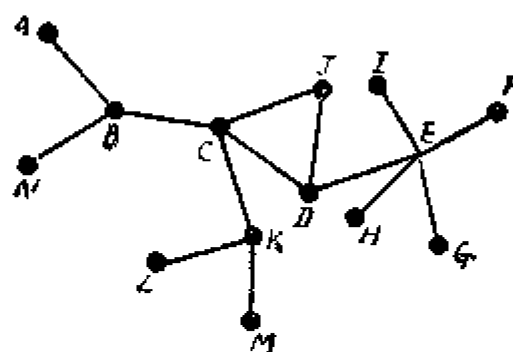


图 12—14

如图12—13和图12—14所示的电信网络只是概括地表示一个电话系统，其实在每一个点所表示的交换局或电话局中，还有一个并不简单的内部连接网络呢。每一个交换局中有很多接线器至少有几万个接点组成几百到几千个接线器组，这些部件之间的互相连接叫做中继。我们可以把交换局间的中继作用用一个图来表示，其中每一个点代表一个接线器组，每一条边代表一组联接线或从一个机键级到另一级之间的一条中继线。

这里我们着重研究中继问题，所以必须先让我们弄清楚一个图和实际网络之间的关系。图12—15是一个二级交换列，左边A接线器的12个输入中的任何一个可以连接到右边B接线器的12个输出中的任何一个。惯常用的中继图如图12—16所示，这里只注意了接线器，而后者的内部接法都是标准方式的。它所对应的完全双向图如图12—17所示，它只注意了互相连接的关系。

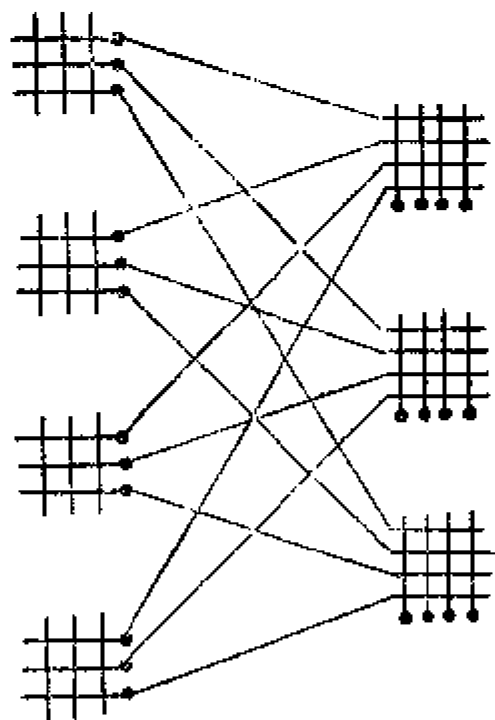


图 12—15

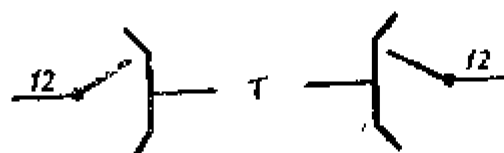


图 12—16

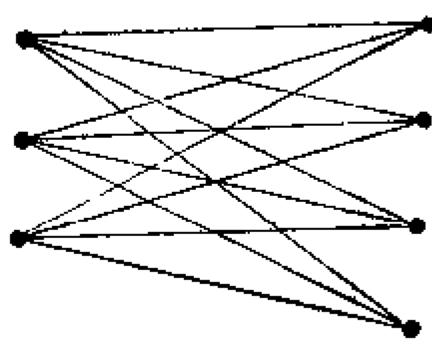


图 12—17

在电话系统的历史中，用户与电话局间最简单的联接方式是由电话局直接放一对线至用户处。用抽象的图来表示，就成为一个星图如图12—18所示。星图也是一棵树，是最简单的树。在电信系统中，星图，双向图和完全图这 3 种图是所有更复杂方式的电信网络的基础。当然对于完全图来讲，由于所用的边或连接线太多，投资费用太高，所以只适用于用户数n很小的场合。

在实际电信网络中，接通一个电话总是要经过几个级联的联接线的。在图论中，我们可以把几个图连起来，每一个图代表一个连接部分。例如有两个小电话局用中继线联通起来，总图就如图12—19所示。从拓扑观点看，m条中继线的末端是否一一对应是无关重要的。所

以图12—19可以简化成图12—20，后者相当于一个电话局有3级接线器A，C，B的中继方式，相邻两级之间有中继线。这种级联方法可以扩展到很多级，譬如说在交换中继里常常可达7级。因此我们可以定义一个n-级图有n个点子组V_i（$i=1,2,\cdots,n$）和（$n-1$）个边子组E_i，分别中继V_i中的那些点和V_{i+1}（$i=1,2,\cdots,n-1$）中的那些点。一般，我们可以假定第一级对应着实际网络中的输入，第n级对应着输出，由输入到输出的正规路径只经过每一边子组的一条边。

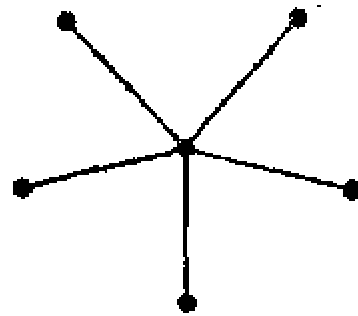


图 12—18

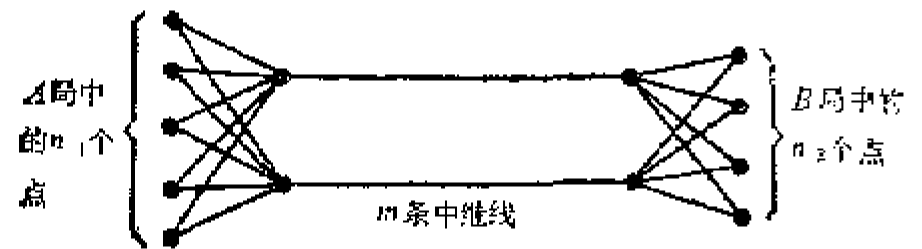


图 12—19

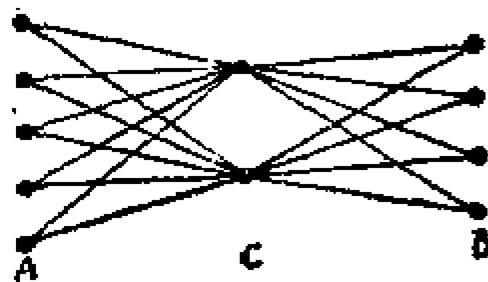


图 12—20

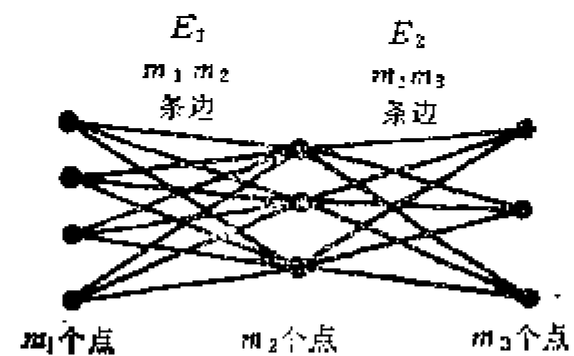


图 12—21

邻接矩阵的运算可以运用在多级图中。设一个矩阵M的元素m_{ij}代表一个独立级的输入i邻接到输出j的，那么这些矩阵的乘积代表那些级的级联接法，每个元素给出一个输入与一个输出之间的路径数。这些路径不包括重新通过的路径段（而在一个简单邻接矩阵的幂乘积中都可能是用的）但是可能包括些共用段的。

图12—21是一个完全n-级图$C(m_1, m_2, \cdots, m_n)$（这里C代表回路），它在第i级有m_i个点（$i=1,2,\cdots,n$）；相邻级i，$i+1$之间由m_im_{i+1}条边相中继。图12—22是一个非完全n-级图。

一个有用的开关网络的一个特性是在任何一个给定的输入和输出之间存在着些变通路径。以图12—22为例，由点A到点S之间的通路图如图12—23中实线所示，这是图12—22的一个子图，它包括由输入点A到输出点S之间的所有路径。如图在V_1中有m_1个点，在V_n中有m_n个点，那么这个网络就有m_1m_n个通路图。

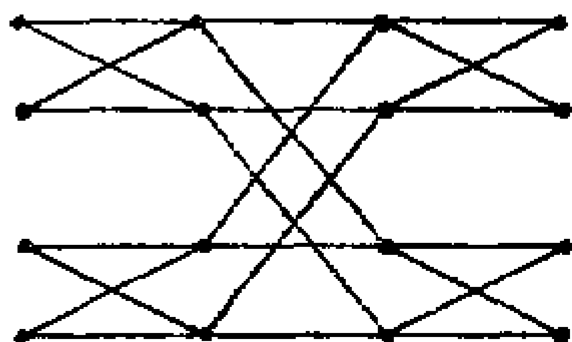


图 12—22

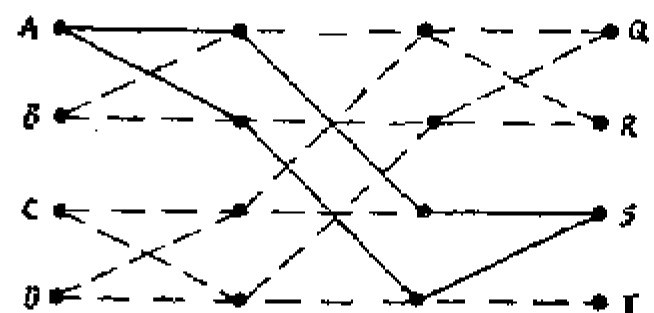


图 12—23

一个三级交换列，如图12—24(a)所示，可用$C(a, c, a)$来表示，它的通路图如12—24(b)所示。很明显，它的连通度为C。在这样一个网络中，当要求接通点A和B时，接通

的概率，如果其它条件一样，则根据 c 的增加而增加；当 c 相当大时，则接通的概率将为1，就是说，这个网络是不会阻塞的。这种情况的条件是 $c \geqslant 2b-1$，其中 b 是一个外部（A或B）接线器所接的输入数。理由是 c 条路径中最多有（$b-1$）条由于接到同一A接线器的其它连接所阻塞；而另有($b-1$)条由于接到同一B接线器的其它连接所阻塞，因此有了($2b-1$)条径路，那么至少有一条是空闲的。有人证明过，较小的连接度，即 $c \geqslant b$ 已经足够用以完成任意接通，只要那些路径可以共同选用而不是独立无关的。这种网络称为重配置网络。

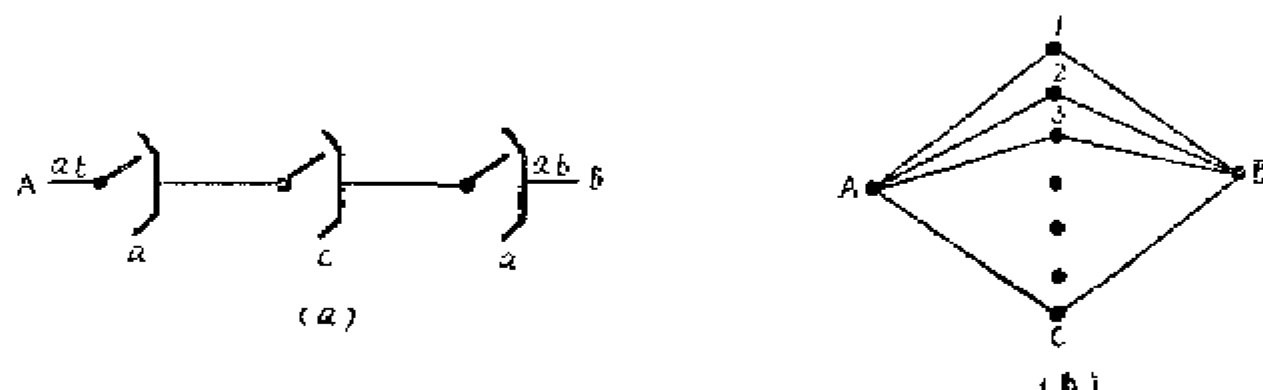


图 12—24

第四节 优先约束树

在通信网络设计中，常常碰到这样的问题，要求联络某些地点，而联接线的建设费用又要求最低，还可能受到一些别的约束条件。某一容量的联接线费用是与它的长度相关的。一个典型的约束条件是各点的业务量不致使联接线过负载。这种通信网络的最经济方案是一棵树，尤其是棵最短生成树，它的求法在第三章第九节中已经介绍过。但是在那里我们只以长度为权而没有其它约束。现在我们讨论除了长度外还有其它约束时求优化约束树的方法。

例如有一个通信网络如图12—25所示，各边所注的权系里程。如果只以里程为权，那么所求的最短生成树将如图12—26所示。假设点v_2是长途电话大区中心局，其它各点v_1，v_3，v_4，v_5为省中心局。再假设省中心局v_5的话务量对于联接线或边（v_3v_5）来讲是太繁重了，那么由v_2，v_3和v_5所组成的分段就不合适。设点v_5和v_4的话务量比边（v_2v_4）和（v_4v_5）的负载量并不太大，那么我们删除边（v_3v_5）而增加边（v_4v_5），新的树就成为如图12—27所示，它的总权数为16，而对图12—26的最短生成树的总权数是 8 。这就说明最短生成树是解决有约束条件的下限。

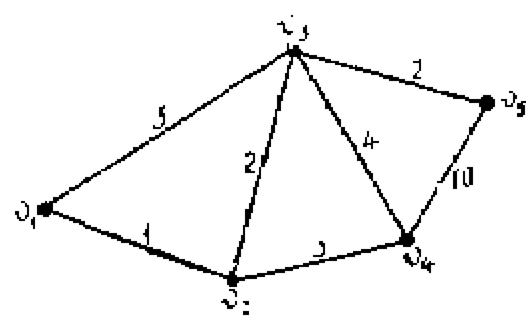


图 12—25

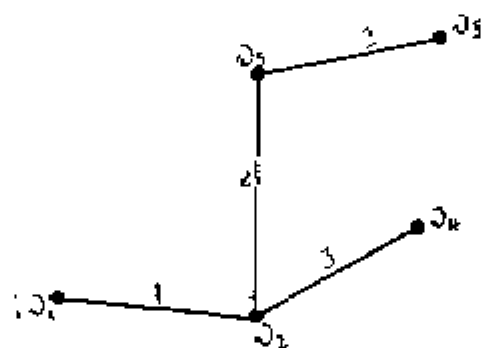


图 12—26

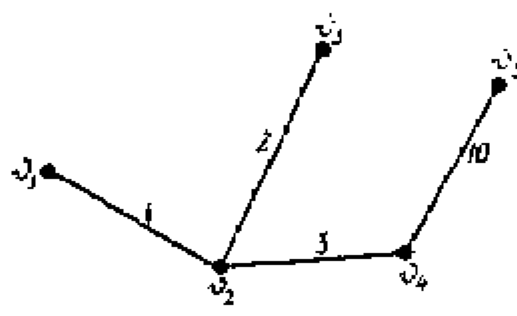


图 12—27

当求最小权树时设有二个约束条件，第一个约束是每一个局 i 的话务量 $m_i \geqslant 0$，并且任何一个分段中各省中心局的总话务量对某一规定局 r 而言不得超过一个规定的最大值M。为了保证有一个可行的解答，我们假定所有省局 i 的$m_i<M$。第二个约束是每一分段的省局数不能多于K。如果$K=1$，那么唯一可实行的方案是星图，就是说每一个省局只能用直达联接线直接连到大区中心局。下面介绍一种算法是一种用直接推断的方法，假定$K \geqslant 2$，这种

算法所得到的结果虽然不一定总是最小权树，但是实际应用起来是很好的。

设假定图的点组N有n个点，点1是长途电话大区中心局，而点2，3，…，n是省局。边的权$w_{ij}(w_{ij}=w_{ji}$，$w_{ii}=0$，$i,j=1,2,3,\cdots,n)$。约束有二：（1）对点1而言，每一分段各省局的话务量m_i的和不许超过某一最大值M；（2）每一分段中省局数不许大于K。现在要求找寻这样的最小权树。在此法中，图用邻接矩阵或接通矩阵来表示。

第一步：首先定义里程l_i，$i=1,2,3,\cdots,n$，作为一组省局点权；置$l_i\leftarrow w_{i1}$。对于一个完全图G，它的点组为N。定义G_i作为包括点i的一部分；置$ci\leftarrow\{i\}$，$i=1,2,3,\cdots,n$。定义t_{ij}，i，$j=1,2,3,\cdots,n$为一组测试值，并置$t_{ij}\leftarrow w_{ij}-l_i=w_{ij}-w_{i1}$。如果在$i$和$j$之间无边连通（即$w_{ij}=\infty$）或如$m_i+m_j>M$，则置$t_{ij}\leftarrow\infty$。置$x_{ij}\leftarrow 0$，$i,j=1,2,3,\cdots,n$。

第二步：找寻可能的新边。找寻

$$t_{i^*j^*}=\min\{t_{ij}\,|\,i,j\text{不在同一部分中}\} \qquad (12-16)$$

如果$t_{i^*j^*}=\infty$，停止。

第三步：在约束条件下测试是否可能再增加边(i^*,j^*)。为了方便起见，新部分用c表示，$c_{i^*}\cup c_{j^*}$，即增加边(i^*,j^*)。如果c'中不包括点1，并且仅当（1）如$\Sigma\{m_i\,|\,i\in c'\}\leqslant M$和（2）$c'$中的省局数小于或等于$K$时，是符合约束条件的。如果$c'$中包括点1，那么仅当（3）由于$c'$的改变或产生的任何一个分段（分段是对点1而言的）中的省局的话务量之和少于或等于M，和（4）在任何一个这样的分段中省局数少于或等于K时，才是符合约束条件的。在上述两种情况时，如果不符合两个约束条件，置$t_{i^*j^*}\leftarrow\infty$，返回第二步。

第四步：增加边(i^*,j^*)，更新点权和t_{ij}，并且组成新的部分c'。置$x_{i^*j^*}=x_{j^*i^*}\leftarrow 1$。对于$i\in c_{i^*}$，置$l_i\leftarrow l_{i^*}$，然后更新$t_{ij}$。就是说，对于$i\in c_{i^*}$，置$t_{ij}\leftarrow w_{ij}-l_i$。置$c_{i^*}=c_{j^*}\leftarrow c'$。返回第二步。

在应用上述算法求优化约束树时，有人有一个经验，为了求解，可以首先从设想一棵假设的星树开始，即从每一个省局有一条直接联接线通到大区中心局，然后设法产生包括有两个或更多的省局的分段来减少星树的权。当然在约束条件下，星树总是一个可行的优化约束树，但是一般它不是一个最小权的解法。

用上法计算优化约束树时，如果约束条件不是上述的（3）每一分段的总话务量M和（2）每一分段的省局数K，那么在第三步中就要相应地修改。

求优化约束树的方法是不少的，有一个方法是从一个图的最短生成树开始，用交换边的方法来适应约束条件。

第五节　边权交通网络中的各种路径

在边权交通网络中，路径问题的主要研究对象是根据一些约束条件找寻某一对指定点之间或者所有点对之间的一条或几条路径。如果不管约束条件，那么上述的路径由于点数的增加而很快地增加的。如果是一个有n个点的完全图，任何一对点之间的路径数接近于$e(n-2)!$，其中$e=2.718\cdots$。在考虑路径的约束时，最多和最先考虑到的是最小权的问题，就是最短路径问题。其次考虑是界权，就是比最小权稍大些的路径。最后是既有权的约束又有共用点或共用边的约束的路径。关于最短路径的算法可参看第十章第六节。

有时，由于某种原因，如交通线路发生故障，不可能再使用最短路径时，只得求其次而

采用比最短路径稍远一些的路径，这就是界权问题。我们可以把最短路径较远些的路径分为第二短，第三短，等等路径，即第 k 最短路径，$k=2$，3，…。在实际上，我们并不计较第几最短而是要求点 i 和 j 之间的距离不要超过一个里程 M，即 $M \geqslant d_{ij}$，后者为最短路径。如果一条路径的里程小于或等于 M，我们称这条路径是可以接受的或容许的。假设由 i 到 k 有一条路径 P_k，它的长度为 l_k，只要 $l_k+d_{kj} \leqslant M$，这就是一条由 i 到 j 包括 P_k 作为一个子组的容许路径。而且这种容许路径不止一条。另一方面，如果 $e_k+d_{kj}>M$，那就不是一条容许路径了。下面是一个在图 $G(V,E)$ 中求点 i 和 j 间所有容许路径和里程的算法。

第一步：开始。这个算法要用到上面提到过的距离矩阵 D。设 m 是点的下标，并且置 $m \leftarrow 1$，和 $i_m \leftarrow i$。设 P 是一条测试路径，并且置 $P \leftarrow \phi$，从空组开始。设 L 是路径的累计长度，并置 $L \leftarrow 0$。

第二步：找寻下一条边。设 i_a 是直接关联到 i_m 的一个点，并且 $P \cap \{(i,i_a)\}$ 不包括回路和 $L+l_{i_m i_a}+d_{i_a j} \leqslant M$。置 $P \leftarrow P \cup \{(i_m,i_a)\}$，并且进行第四步。如果在点 x 和 y 之间有一条边，点 x 和点 y 是直接相连的，就是 $e_{xy}<\infty$。如果所用直接连通到 i_m 的边都已经测试过，那么进行第三步。

第三步：算法完了或回溯。如果 $m=1$，停止。否则置 $m \leftarrow m-1$，返回第二步。

第四步：测试整个路径或继续下去。如果 $i_a=j$，保留 P 和 L，置 $P \leftarrow \phi$ 和 $L \leftarrow 0$，返回第二步。所谓保留 P 和 L，在计算机术语中的意思是把组成 P 的那些边和 L 值的清单存储在存储器中，以便用作结果的一部分。

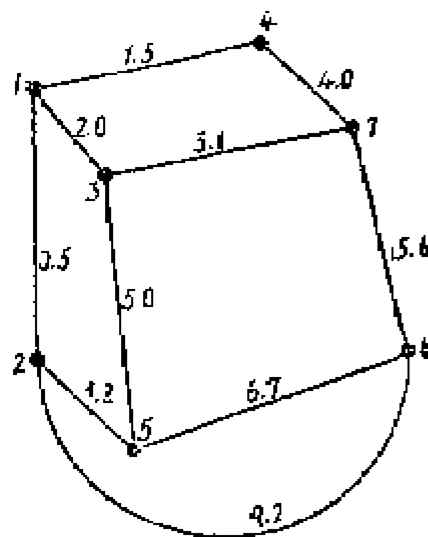


图 12—28

例如，在图12—28中，各边上所注的数字系相邻两点之间的距离，我们规定 $M=7$，$i=1$，$j=5$。如果在第二步，我们试以 $i_a=2$，就是说在路径 P 的第一条边用（1，2），我们从矩阵 $D^{(7)}$ 有 $L+l_{12}+d_{25}=0+0.5+1.2=1.7 \leqslant M$。因此边（1，2）是试算路径的第一条边。在第四步我们得 $L=0.5$，$m=2$ 和 $i_2=2$。当我们再进行下去时，我们发现直接连通到点2的，除点1外，还有两个点，即点5和点6。前者只要再加一条边（2，5），成为一条容许（而且也是最短）路径 $P_{15}=\{1,2,5\}=\{(1,2),(2,5)\}$；而后者要加两条边（2，6）和（6，5），因此 $l_{12}+l_{26}+S_{65}=0.5+9.2+6.7>M$，这是超过了 M，是不容许的。

继续用这个例子，如果我们试 $i_a=3$ 作为直接连通到点1的后继点，我们在第二步又有 $m=1$，和 $L=0$。因此 $L+l_{13}+d_{35}=0+2,0+3,7 \leqslant M$，所以边（1，3）是试算路径 P 的第一条边。这不是从点3到点5的最短路径，就是说现在用了边组{(3，1)，(1，2)，(2，5)}，而不是边{(3，5)}，而经由点3的容许路径是{(1，3)，(3，5)}。直接连通到点1的最后一个点是点4，我们有 $L+l_{14}+d_{45}=0+1,5+3,2 \leqslant M$ 不过经由点4实际上是没有从点1到点5的容许路径的。

容许路径算法有一个缺点，就是容易出差错，如像上面例子中所指出的。不过这个算法比较简单，也相当有效。有一点应该注意的是所得的路径不是按长度系统地产生的。所以在算完后应该依照长度或别的次序列出一个路径表，以便比较选择。

在交通网络中也常常会碰到这样的情况，如果点 i 和 j 之间路径经由一个或几个中间

点，而当任何一个中间点发生损坏或不空闲时，信息就不能通过。同样，如果几条路径要通过一条或几条共用边时，任何一条共用边发生故障就影响各条路径的交通。所以在点 i 和 j 之间的交通路径，除点 i 和 j 外，我们要找寻点不共用路径或边不共用路径。两条点不共用路径一定是边不共用的，但是反过来就不一定正确。

一般地讲，在交通网络中，我们非但要求非共用性，而且也常有别的约束。例如，如果我们在点 i 和 j 之间只需要两条路径，我们可能希望使用（1）最短路径，譬如说 P_s 和(2)最短路径并且是与路径 P_s 点不共用的，譬如说 P_d。关于最短路径 P_s 的求法，我们在上面已经讲过。下面介绍路径 P_d 的一种求法。

如果有一个交通网络 $G(V,E)$ 如图12—29所示，从点 1 到点 4 之间的最短路径为 $P_s=\{(1,6),(6,5),(5,4)\}$。现在要求找寻一条与路径 P_s 点不共用的最短路径 P_d。先把图12—29中 P_s 所经由的所有中间点 6 和 5 删除得一新图 $G'(V',E')$，如图12—30 所示。从 G' 中，得知点 1 和 4 之间的路径有二：一为 $\{(1,2),(2,3),(3,4)\}$，二为 $\{(1,2),(2,4)\}$，这两者中的较短者即为所要求的 P_d。如果这个网络许可的话，我们还可以得到与 P_s 和 P_d 都是点不共用的最短路径 P_f。当然在这个网络中是不能有再与点 1 与 4 之间的第三条点不共用的路径了。这里也可以看出，如果在图12—29 中原先是求点 7 与其它点之间的最短路径 P_s，那么因为点 2 将是一个中间点，后者的删除就不能再得其它点不共用的路径了。

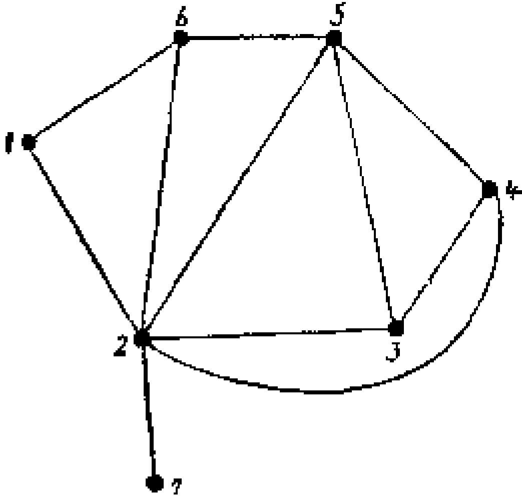


图 12—29

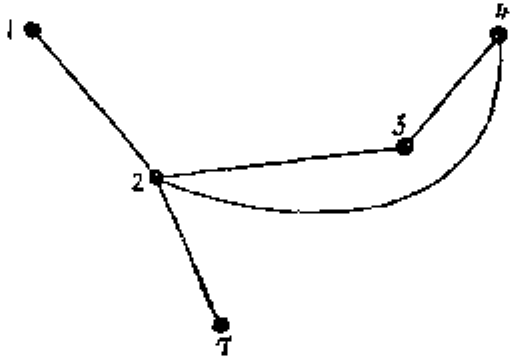


图 12—30

第六节 电视接力网络

电视（或微波）接力网络的修建费用主要决定于接力站间的距离和接力站场所的地形。由于电视波（或微波）的传播可以指向任何方向，而且不需要电波反射回来，所以如果由若干个城市的电视台组成一个电视接力网络，这个图是一个无向图。

例如有八个城市1，2，3，4，5，6，7，8 的电视台组成的一个电视接力网络，其中台 1 为主要发射台，如图12—31所示。图中每一可能接力线路的建设费用，如只算高位数的款项位将如表12—1 所示。

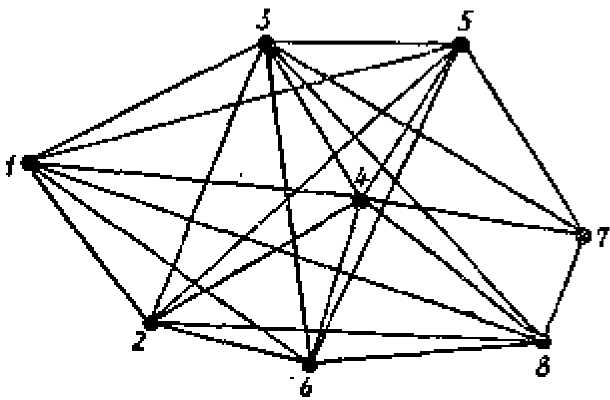


图 12—31

由表12—1 可见，由台 4 到台 7 的建设费用等于 5 万元。如果把图12—31的邻接矩阵中的元素值代以建设费用，那么这种矩阵称为费用矩阵。如用矩阵方式表示，则与表12—1 相

似。费用矩阵中的元素值如为∞，则表示这两台之间的转播由于技术上的困难而不能实现。我们现在的问题是如何使接力网络的建设费用为最小。

建　设　费　用　表　　　　表12—1

费用单位(万元) 城市＼城市	1	2	3	4	5	6	7	8	费用单位(万元) 城市＼城市	1	2	3	4	5	6	7	8
1	0	1	3	3	5	2	∞	6	5	5	5	1	3	0	6	5	∞
2	1	0	4	3	5	2	∞	5	6	2	2	6	4	6	0	8	4
3	3	4	0	3	1	6	7	9	7	∞	∞	7	5	5	8	0	9
4	3	3	3	0	3	4	5	6	8	6	5	9	6	∞	4	9	0

这个问题在图论中就是求图12—31中以台 1 为根的最短生成树。我们已经在第三章第九节中介绍过两种算法。现在把图 3 —19的正式算法叙述如下。

这种算法的基本原理是一个有几个点的连通图的 $G(V,E)$ 的最短生成树有一个唯一的解T°，其中T°是一个边组$\{e_1,e_2,\cdots,e_{n-1}\}$，后者产生如下：

e_1是E中最小权的边；

e_2是$E\ominus\{e_1\}$中的最小权的边，并且它不会与边组$\{e_1\}$形成一个回路；

e_r是$E\ominus\{e_1,e_2\cdots e_{r-1}\}$中的最小权的边，并且它不会与边组$\{e_1,e_2,\cdots,e_{r-1}\}$ 形成一个回路。

这个算法要利用与表12— 1 相似的费用矩阵。假设称图12—31的费用矩阵为矩阵K，

第一步：将矩阵K中对角线以上的各元素值按递增次序排列，称为组P。开始$T\leftarrow 0$和标号$L=0$。

第二步：设l_{ij}是P中的第一个元素。置$t_{ij}\leftarrow 1$。标号点 i 和点 j 为 1。

第三步：对于$L_i=L_j\neq 0$ 的每一点对 i 和 j 从P中删除l_{ij}。

第四步：设l_{ij} 是P中的第一个元素。置$t_{ij}\leftarrow 1$。检查点 i 和 j 的标号。如果两者之中的一个标号为 0，则把它标以与另一个点相同的标号。如果两者中没有一个是标 0 的，则把 P 中的所有较高的标号改标成较低的标号。如果两者都为 0，则把它们标成比新近标最高标号的更高号的。

第五步：如果在T 中的正元素数等于$n-1$，因为T 已经给出了所需要的树，就停止。其次，如果组P是空的，表示不存在生成树，就停止。否则返回第三步。

让我们把这个算法应用于图12—31来找出以台 1 为根的最短生成树。

第一步：$T=0$

$L=\{0,0,0,0,0,0,0,0\}$

$P=\{l_{12},l_{35},l_{16},l_{26},l_{13},l_{14},l_{24},l_{34}$

$l_{45},l_{23},l_{46},l_{68},l_{15},l_{25},l_{28},l_{47},l_{57},l_{18}$

$l_{36},l_{48},l_{56},l_{37},l_{67},l_{38},l_{78}\}$

第二步：置$t_{12}=1$

$L=\{1,1,0,0,0,0,0,0\}$

第三步：从P中删除l_{12}。

第四步：置$t_{35}=1$。因为点 3 和 5 的标号都是 0，所以这两点都标以 2

$L=\{1,1,2,0,2,0,0,0\}$

第三步：从P中删除l_{35}。

第四步：置$t_{16}=1$。因为点1和6中的点1标号为1而点6标号为0，所以把点6标以与点1相同的标号1。

$L=\{1,1,2,0,2,1,0,0\}$

到此，对于标号=1的，从P中删除l_{12}，l_{16}，l_{26}，由于已经删去了l_{12}，这次应删除l_{16},和l_{26}；对于标号=2的，l_{35}已经删除了。

第三步：$P=\{l_{13},l_{14},l_{24},l_{34},l_{45},l_{23},l_{46},l_{68},l_{15},l_{25},l_{28},l_{47},l_{57},l_{18},l_{36},l_{48},l_{56},l_{37},l_{67},l_{38},l_{78}\}$

第四步：置$t_{13}=1$因为点1，3的标号都不是0所以把P中的2都改标为1。

$L=\{1,1,1,0,1,1,0,0\}$

第三步：从P中删除l_{13}，l_{15},l_{23}，l_{25}，l_{36}和l_{56}。

$P=\{l_{14},l_{24},l_{34},l_{45},l_{46},l_{68},l_{28},l_{47},l_{57},l_{18},l_{48}l_{37},l_{67},l_{38},l_{78}\}$,

第四步：置$t_{14}=1$。在L中改标点4为1。

$L=\{1,1,1,1,1,1,0,0\}$

第三步：从P中删除l_{14}，l_{24}，l_{34}，l_{45}和l_{46}

$P=\{l_{68},l_{28},l_{47},l_{57},l_{18},l_{48},l_{37},l_{67},l_{38},l_{78}\}$

第四步：置$t_{68}=1$。在L中改标点8为1。

$L=\{1,1,1,1,1,1,0,1\}$

第三步：从P中删除l_{18}，l_{28}，l_{38}，l_{48}，l_{68}

$P=\{l_{47},l_{57},l_{37},l_{67},l_{78}\}$

第四步：置$t_{47}=1$。在L中改标点7为1。

$L=\{1,1,1,1,1,1,1,1\}$

第五步：因为T中有了7个正元素，算法终止。

所得的最短生成树的树枝为t_{12}，t_{35}，t_{16}，t_{13}，t_{14}，t_{68}，和t_{47}，组成的最短生成树如图12—32所示。这个网络的最低建设费用为

$$1+1+2+3+3+4+5=19$$

在这个例子中，如果我们在开始的第一步中，把边组P排列改为

$$P=\{l_{12},l_{35},l_{26},l_{16},l_{13},\cdots\cdots\}$$

则照算法进行下去所得的最短生成树如图12—33所示，所得的最低建设费用也是19。这是因为段（1，6）和段（2，6）的建设费用是相等的。

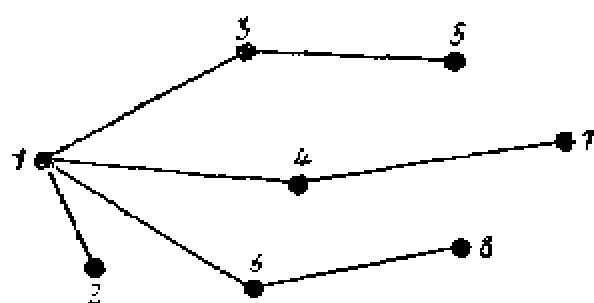


图 12—32

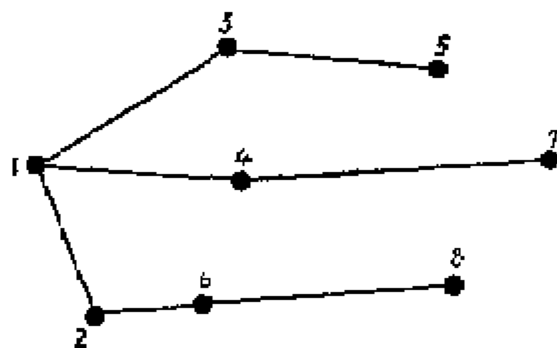


图 12—33

第七节　有向边权交通网络

当一个边权交通网络中每边有一个定向表示通过边的流量只沿规定方向时，这个网络就叫做有向边权交通网络。对一条路径$P_{r_{ij}}$从点i到j分配流量$\psi_{r_{ij}}$时，除了路径必须是从i到j的一条有向路径外，其它是与无向边权交通网络一样的。同样，对一个有向边权交通网络分配流量ψ_{ij}时是与无向边权交通网络一样的，从i到j的极限负载量是i到j所能分配的最大流量。我们应该注意在有向边权交通网络中，极限负载量t_{ij}不一定等于极限负载量t_{ji}。

为了讨论有向边权交通网络的特性，这里提出半截割的术语，它的定义是：

设$S_{ij}=E(V_i\times\overline{V}_i)\cup E(\overline{V}_i\times V_i)$是一个有向图中的一个截割组，其中$i\in V_i$，$j\in\overline{V}_i$，那么$S_{ij}$的半截割$s_{ij}$和$s_{ji}$是

$$s_{ij}=E(V_i\times\overline{V}_i) \tag{12—17}$$

$$s_{ji}=E(\overline{V}_i\times V_i) \tag{12—18}$$

注意s_{ij}和s_{ji}是些边不共用组，而且

$$s_{ij}\cup s_{ji}=S_{ij} \tag{12—19}$$

象在无向边权交通网络那样，在一个有向边权交通网络中的一条边，如果它的边流量等于它的边负载量，这条边就是饱和了。可是在一个有向边权交通网络中，关于一个分配流量ψ_{ij}的一个基础饱和截割组的定义是与无向边权交通网络中的有些不同。在一个有向边权交通网络中，如果截割组S_{ij}的半截割组s_{ij}中每条边是饱和的，并且在S_{ij}的半截割s_{ji}中没有一条边有非零流量的话，那么在一个从i到j的分配流量ψ_{ij}条件下，这个截割组S_{ij}是一个基础饱和截割组。

例如，如果我们分配流量ψ_{ij}如图12—34所示，那么在截割组$S_{ij}=(a,b,c,d)$中各边是饱和的，可是，因为在S_{ij}的半截割s_{ij}和s_{ji}是

$$s_{ij}=(a,b,d)$$

$$s_{ji}=(c)$$

而且分配给边c的非零流量$\psi(c)$，所以S_{ij}不是一个基础饱和截割组。

当我们分配流量ψ_{ij}如图12—35所示时，截割组$S_{ij}=(a,b,c,d)$变成一个基础饱和截割组。有了这些定义，我们可以提出下一定理12—4。

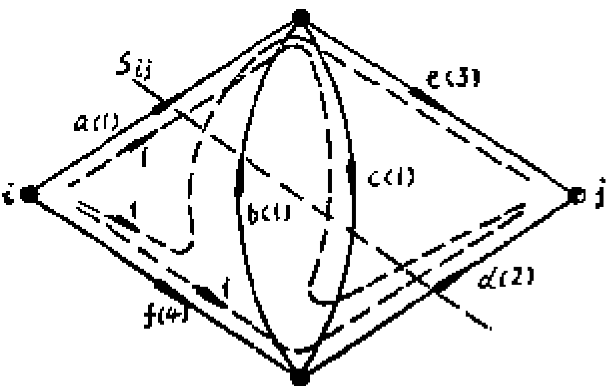


图　12—34

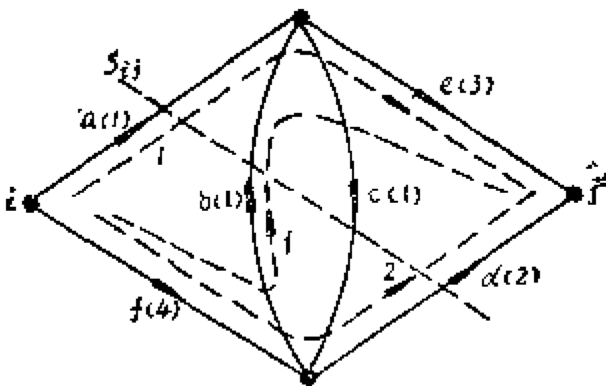


图　12—35

定理12—4：当且仅当有一个分开i和j的基础饱和截割组时，分配给一个有向边权交通网络的流量ψ_{ij}是最大值。

这个定理的证明与定理12—2的证明正好相同，只不过我们用有向边权交通网络的一些基础饱和截割组。我们可以知道对于分配最大流量ψ_{ij}的每一个可能的方法，至少有一个截

割组是分开 i 和 j 的基础饱和截割组。这个截割组叫做极限负载量 t_{ij} 的对应截割组。

由这个定理12—4我们得另一个定理。

定理12—5、极限负载量 t_{ij} 等于

$$t_{ij}=\min\{\bigcup(s_{p_{ij}})\}\ S_{p_{ij}}\text{的半截割}$$
$$s_{p_{ij}}\in\{S(i,j)\} \tag{12—20}$$

例12—3 在图12—36中的有向边权交通网络 G 中，$\{S(i,j)\}$ 包括下列分开 i 和 j 的截割组：

$$S_{1_{ij}}=(a,g,h)$$
$$S_{2_{ij}}=(a,b,c,d)$$
$$S_{3_{ij}}=(d,e,f)$$
$$S_{4_{ij}}=(b,c,e,f,g,h)$$

这些截割组的半截割 $s_{p_{ij}}$ 是

$s_{1_{ij}}=(a,h)$

$s_{2_{ij}}=(a,b,d)$

$s_{3_{ij}}=(d,e)$

$s_{4_{ij}}=(c,e,h)$

所以 $t_{ij}=\min\{\bigcup(s_{p_{ij}})\}=\min\{5,4,5,9\}=4$

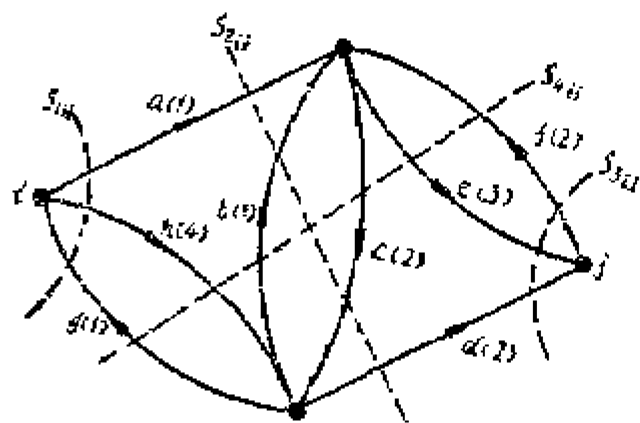


图 12—36

另一方面

$s_{1_{ji}}=(g)$，$s_{2_{ji}}=(c)$，

$s_{3_{ji}}=(f)$，$s_{4_{ji}}=(b,f,g)$

所以

$$t_{ji}=\min\{U(s_{p_{ji}})\}=\min\{1,2,2,4\}=1$$

第八节 损耗边权交通网络

有几种具体系统，不是所有的流量都能够达到它们的目的地的。一个原因是在系统中可能有缺点，如象管道中的漏泄。为了表达这些系统，我们规定一个损耗有向边权交通网络的每一条边有一个边效率 α_p，其中 $0\leqslant\alpha_p\leqslant 1$。为了理解边效率 α_p 起见，假设边 e_p 连接在点 k 和 m 之间，它的定向是从 k 到 m。那么当流量 ψ 从点 k 进入 e_p 时，通过 e_p 到达点 m 的流量是 $\alpha_p\psi$，所以流量 $(1-\alpha_p)\ \psi$ 是可以看做是边 e_p 中损失掉的。

为了讨论损耗边权交通网络的特点起见，我们对半截割 s 的两个值 $\bigcup[s(c)]$ 和 $\bigcup[s(\alpha_c)]$ 定义如下：

$$\bigcup[s(c)]=\sum_{e_p\in s}c_p \tag{12—21}$$

$$\bigcup[s(\alpha c)]=\sum_{e_p\in s}\alpha_p c_p \tag{12—22}$$

式中 α_p 和 c_p 分别为边 e_p 的边效率和边负载量。

因为流量的损失发生在边中，我们对一个损耗边权交通网络 G 分配流量 ψ_{ij} 如下，设

$P_{r_{ij}}$是G中从点i到点j的一个有向路径，它的行走是$(e_1, e_2 \cdots e_n)$。那么分配一个流量$\psi_{r_{ij}}$给路径$P_{r_{ij}}$的意思是对边$e_p(p=1,2,\cdots,n)$的边流量所增加流量$\psi'(e_p)$为

$$\psi'(e_1)=\psi_{r_{ij}}$$

$$\psi'(e_2)=\alpha_1\psi'(e_1), \cdots \psi'(e_n)=\alpha_{n-1}\psi'(e_{n-1}) \qquad (12-23)$$

有一个条件就是$\psi'(e_p)$必须满足

$$\psi'(e_p) \leqslant c_p - \psi_0(e_p) \qquad (12-24)$$

其中$\psi_0(e_p)$是早先分配给边e_p的流量，$p=1,2,\cdots,n$。因此分配给边e_p的总流量成为$\psi'(e_p)+\psi_0(e_p)$。

设$P_{1_{ij}}, P_{2_{ij}}, \cdots, P_{m_{ij}}$是$G$中从点$i$到点$j$之间所有可能的有向路径。那么为了分配这个总流量$\psi_{ij}=\sum_{r=1}^{m}\psi_{r_{ij}}$，我们可以分别给$P_{1_{ij}}$，$P_{2_{ij}} \cdots$，$P_{m_{ij}}$分配流量$\psi_{1_{ij}}$，$\psi_{2_{ij}} \cdots$，$\psi_{m_{ij}}$。很清楚，分配给$G$的任何流量$\psi_{ij}$可以分散成另外流量。所以我们可以证明我们能够增加从i到j的流量一直到有一个分开点i和点j的基础饱和截割组为止。

在无损边权交通网络的情况下，当我们有一个基础饱和截割组S_{ij}时，流量ψ_{ij}总是最大量。可是在另一方面，如果这个边权交通网络是有损耗的，分配流量ψ_{ij}即使会产生一个基础饱和截割组，流量可能不是最大量。这可以从下面的例子中容易看出来的。例如在图12—37中所示的损耗边权交通网络，其中每条边旁括弧中的第一个数字是边负载量，而第二个数字是边效率。分配流量如图所示，截割组$S_{ij}=(c)$成为一个基础饱和截割组。可是，因为从点i到j可以如图12—38所示分配最大的流量，所以进入点i和离开点j的流量不是最大量。

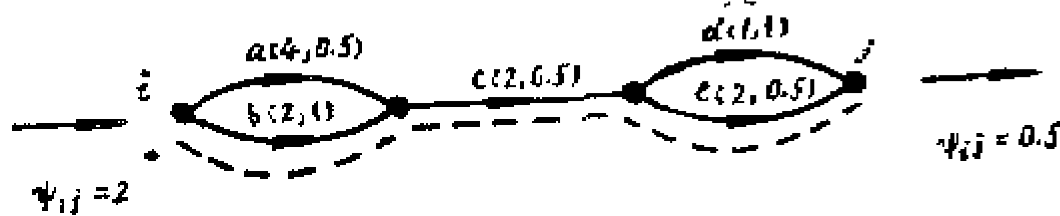


图 12—37

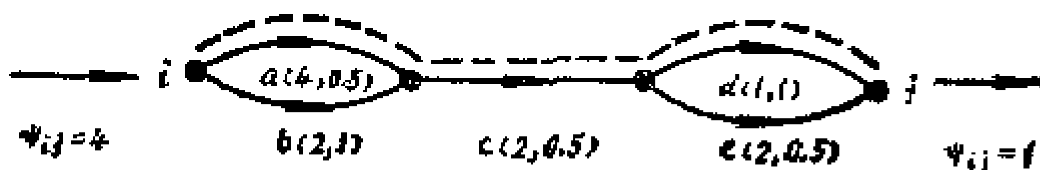


图 12—38

这个例子表明我们可以有许多方案流量ψ_{ij}；当分配这些方案流量时，我们一定会有些基础饱和截割组S_{ij}的。而且当分配流量ψ_{ij}时，进入点i的流量是与离开点j的流量是不同的。所以损耗边权交通网络的极限负载量的定义是与那些无损边权交通网络的有所不同的。现在介绍损耗边权交通网络中所用的一些定义如下。

为了在点j得到最大的流量起见，对一个网络G所能分配的从i到j的最大流量ψ_{ij}叫做源点极限负载量，符号为$\overline{t}_{ij}$。当分配最大流量ψ_{ij}时，在点j所将收到的最大流量叫做收点极限负载量，符号为$\underline{t}_{ij}$。

必须注意到网络中有一个截割组S_{ij}。当分配任何流量ψ_{ij}至少会产生一个基础饱和截割组的。将使S_{ij}成为一个基础饱和截割组。这个截割组叫做这一流量的对应截割组。因为当分配最大流量ψ_{ij}时，S_{ij}显然是一个基础饱和截割组，所以S_{ij}是源点和收点极限负载量$\overline{t}_{ij}$和$\underline{t}_{ij}$的对应截割组。有了这个定义以后，我们有下面另一定理。

定理12—16：对于一个损耗有向边权变通网络中任何点 i，j 和 k，或者

$$\underline{t}_{ij} \geqslant \underline{t}_{kj} \tag{12—25}$$

或者

$$\overline{t}_{ij} \geqslant \overline{t}_{ik} \tag{12—26}$$

证：设 $S_{ij}=E(V_1\times\overline{V_1})\cup E(\overline{V_1}\times V_1)$ 是 $\overline{t}_{ij}$ 和 $\underline{t}_{ij}$ 的对应截割组，其中 $i\in V_1$，$j\in\overline{V}_1$。再设 $S_{ik}=E(V_2\times\overline{V}_2)\cup E(\overline{V}_2\times V_2)$ 是 $\underline{t}_{kj}$ 的一个对应截割组，其中 $k\in V_3, j\in\overline{V}_3$。

情况一，点 k 是在 V_1 之中，根据 $j\in\overline{V}_2$ 或 $j\in V_2$，又有两种情况。

（一）当 $j\in\overline{V}_2$ 时，或者 $i\in\overline{V}_3$，如图12—39所示，对于最大流量 ψ_{ij}，S_{kj} 和 S_{ij} 一定是基础饱和截割组因此

$$\underline{t}_{ij}=\underline{t}_{kj} \tag{12—27}$$

对于 $i\in V$，有最大流量 ψ_{kj} 时，S_{ij} 可能是，也可能不是一个基础饱和截割组。因此

$$\underline{t}_{ij}\geqslant\underline{t}_{kj} \tag{12—28}$$

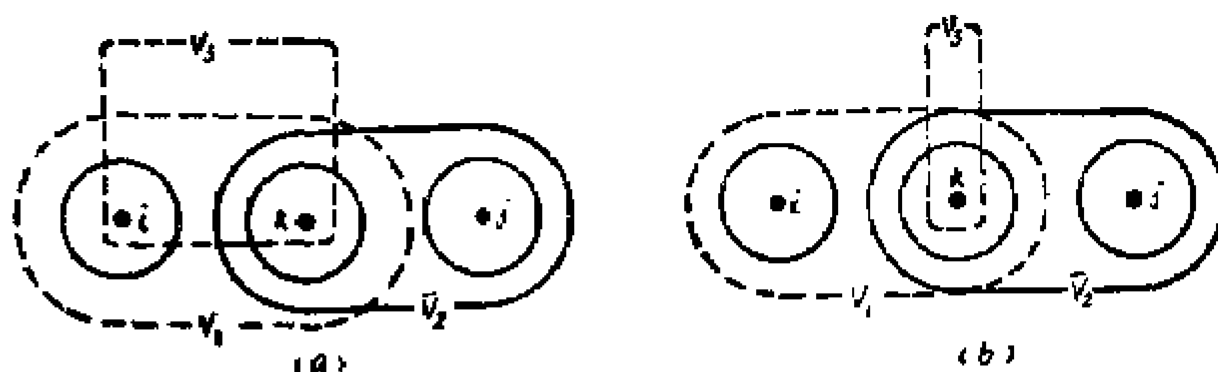


图 12—39

（二）当 $j\in V_2$ 时，情形要看是否 $i\in V_3$ 或 $i\in\overline{V}_3$ 而定，如图12—40所示。对于 $i\in V_3$，在最大流量 ψ_{kj} 时，S_{ij} 一定是一个基础饱和截割组，并且

$$\underline{t}_{ij}=\underline{t}_{kj} \tag{12—29}$$

对于 $i\in\overline{V}_3$，在 ψ_{kj} 时，S_{ij} 不一定是一个基础饱和截割组。所以

$$\underline{t}_{ij}\geqslant\underline{t}_{kj} \tag{12—30}$$

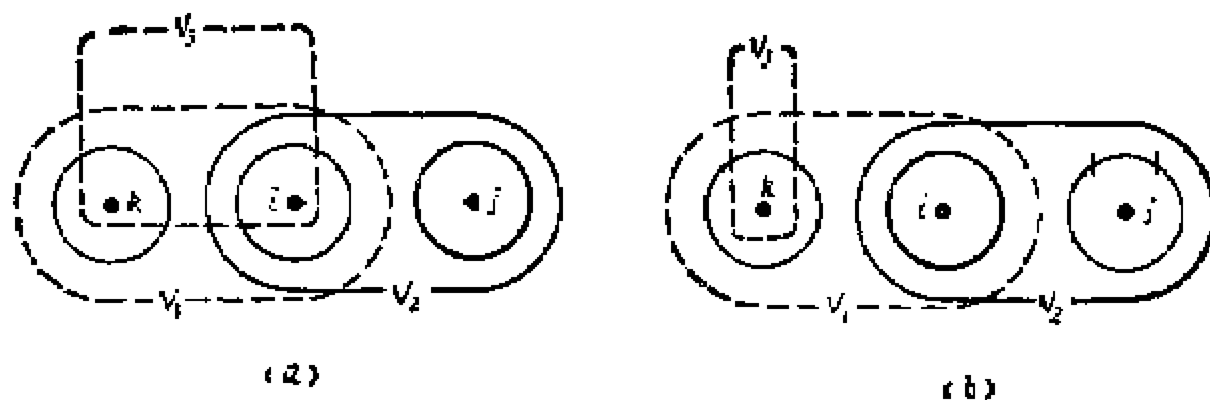


图 12—40

情况二　点 k 是在 $\overline{V}_1$ 之中。我们只需要考虑当 $\overline{V}_2\subset\overline{V}_1$ 的情形。因此 $j\in V_2$，或者 $j\in\overline{V}_2$，假使 $j\in V_2$ 如图12—41所示，那么在 ψ_{ik} 时，S_{ij} 不一定是一个基础饱和截割组因此

$$\overline{t}_{ij}\geqslant\overline{t}_{ik} \tag{12—31}$$

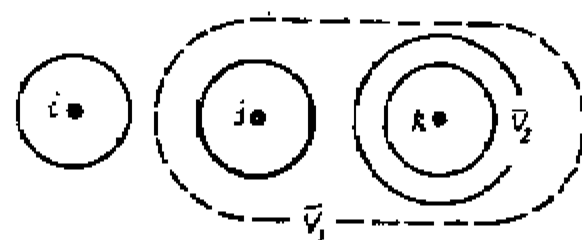


图 12—41

如果 $j \in \overline{V}_2$，在ψ_{1k}时，S_{ij}一定是一个基础饱和截割组。因此

$$\overline{t}_{ij} = \overline{t}_{1k} \tag{12—32}$$

在这一节中所给出的每条边显然也适用于无向损耗边权交通网络。可是，在定义方面必须加以若干修改，例如，对于一个基础饱和截割组就应予修改。

习　　题

12—1　如果分配给图12—42网络的流量$\psi_{12}=6$，试进行分配。

12—2　如果分配给图12—43网络的流量$\psi_{12}=7$，试进行分配。

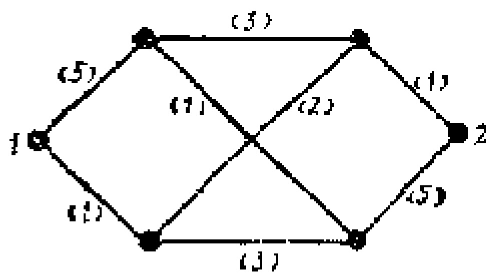


图　12—42

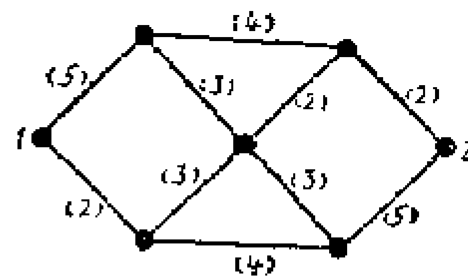


图　12—43

12—3　试求图12—44网络中由1到2的最大流量ψ_{12}。

12—4　试求图12—45网络由点1到2的最大流量ψ_{12}。

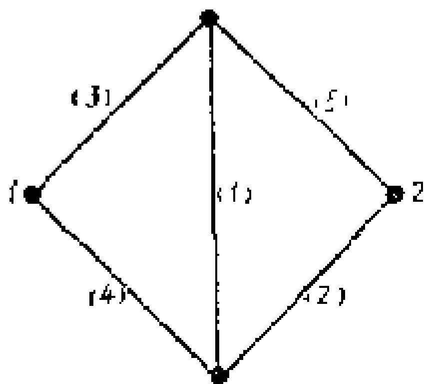


图　12—44

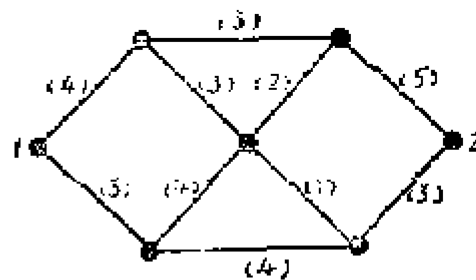


图　12—45

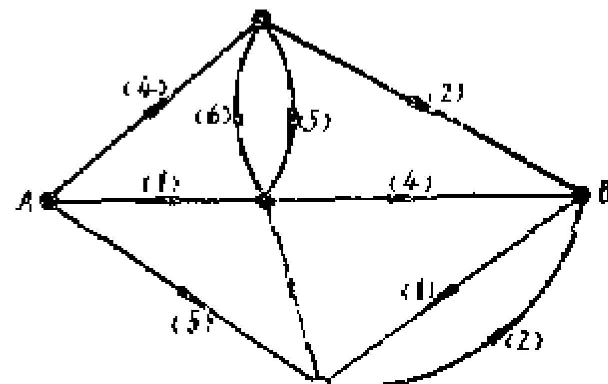


图　12—46

12—5　有一个铁路运输网络如图12—46所示，各边上所注的数字为该边的极限负载量。求由A站到B站之间的最大运输量。

12—6　在图12—47中点1为大区中心局，点2，3，4，5，分别为省局，各省局点旁括号中的数字为各局的话务量，即$m_2=35$，$m_3=20$，$m_4=50$，$m_5=20$，各边上所注的数字为里程，这个系统中有两个约束，即$M=60$和$K=3$。求这系统的优化约束树。

12—7　有一地区附近城市共有6个发电厂。为了互相支援电力供应，拟将这些发电厂之间建成一个供电网。设各厂之间输电线的建设费用如图12—48所示。试问这个供电网应怎样连通法，使供电网的建设费用最为经济？

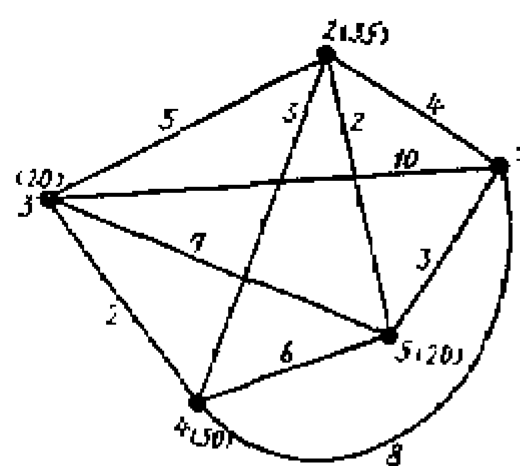


图　12—47

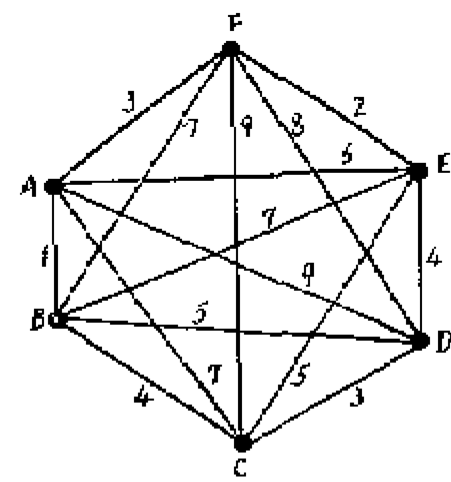


图　12—48

第十三章　点权交通网络

第一节　无向点权交通网络的流量

在铁路运输网络中，不仅有乘降旅客的客运站和装卸货物的货运站，还有解体和编组列车的编组站和区段站。一般地讲，铁路运输网络中的运量，常常受到区间通过能力的限制。就是受每昼夜所能通过的列车对数的限制。上一章中所讲的边负载量和极限负载量就是指区间的通过能力。

但是在铁路运输网络中也常常有这样的现象，区间通过能力能够满足运量的要求，而列车或车辆受阻于某一个或某些编组站或区段站中。这是因为每一列货车所运送的货物往往不是专为发往某一站或某一地点的，而需要在中途某些站将货物列车解体后再重新编组成发往不同方向的货物列车，这就是编组站或区段站的解体作业。所以一个车站的通过能力，尤其是编组站和区段站的通过能力，就指每昼夜通过这个站的列车数或车辆数。铁路运输的运量在很大程度上受着编组站或区段站编解能力的限制。

当铁路运输网络用一个线性图来代表时，非但每条边有一个负载量，而且每个点也有一个负载量，后者叫做点负载量，表示通过点的流量的上限。这样一个线性图叫做一个点权交通网络，在一个点权交通网络中的每条边表示两站之间的传输线。如果这些边是无向的，流量可以沿任何方向通过这些边的。当所有边都是无向时，这个网络就叫做无向点权交通网络。图13—1是一个无向点权交通网络的一个例子，每个点的点负载量表明于每个括弧中。

在通信网络中也是这样，不仅有边权的问题，如上一章所讲的，也有点权的问题。

例如在一个典型数据通信系统的一个站Z中，每一个站有许多输入和输出端机的。当站Z的输入端机处有若干数据进入时，这些端机适当地收集这些数据。如果数据是编码的，它们可能在站Z中译码，然后把这些数据加以变更。例如，如果站Z是一个中间站，它可以提供外加信息；而且某一个进入数据在送到另外一个站去之前需要这样的外加信息，那么所需要的信息就在站Z插入数据中。当一些数据的目的地不同时，它们就分或若干组，这样使得在一组中的数据可以发送到站Z的同一输出端机中去。如果需要的话，这些数据加以编码暂时存储在站Z中，等待有机会发送到次站去。

象在第十二章中所讨论过的边权交通网络一样，一个流量是通过一条路径来分配的。就是说，如果对于一条路径P_{ij}中的所有点，一个流量ψ_{ij}能满足下面关系

$$\psi_{ij} \leqslant C_v - \psi_0(v) \qquad (13—1)$$

其中$\psi_0(v)$是早已分配给点v的流量，那么我们可以分配给P_{ij}从i到j这个流量ψ_{ij}。例如，我们可以如下分配给图13—1中的点权交通网络流量ψ_{ij}。

首先，我们对路径$P_{1_{ij}}=(e_1, e_4, e_6, e_8)$分配$\psi_{ij}=1$。因为

$$1 \leqslant \min\{c_i, c_a, c_d, c_f, c_j\}$$

式中c_i是点i的点负载量，c_a是点a的点负载量等等，所以这个流量$\psi_{1_{ij}}$是可行的。当我们分

配$\psi_{1_{ij}}$时，我们有

$$\psi_0(i)=\psi_0(a)=\psi_0(d)=\psi(f)=\psi(j)=1$$

而

$$\psi_0(b)=0$$

其次，我们对路径 $P_{2_{ij}}=(e_6,e_7,e_8)$ 分配$\psi_{2_{ij}}=1$。因为

$$1\leqslant\min\{c_i-\psi_0(i),\ c_b-\psi_0(b),\ c_d-\psi_0(d),\ c_j-\psi_0(j)\}$$

图 13—1

图 13—2

所以这也是可行的，结果如图13—2 所示。我们可以看出从 i 到 j 不能再分配流量了，所以所能分配的总流量为 2。如果我们另行作不同的分配，我们可以增总流量为 3，分配法如图13—3 所示

在边权交通网络中，如果我们知道截割组，我们就可以计算最大流量。同样在点权交通网络中，如果我们知道点截割，我们可以确定最大流量。为了对点截割下定义，我们就必须先解释从一个线性图中删除一些点的意义。

图 13—3

图 13—4

设V是一个线性图G中的一个点组，所谓在V中删除所有点就是说从G中删除在V中的所有点，并且删除关联到这些点的所有边。例如，如果我们从图13—1 的线性图中删除点 a 和 d，结果图就不再有点 a 和 d 外，也不再有边e_1,e_2,e_4,e_5,e_7和e_8了，象图13—4 所示。

现在对点截割下定义，假使在一个线性图中从点 i 到点 j 至少有一条路径，那么分开 i 和 j 的一个点截割是一个最小点组，从这个点组中删除所有点就破坏了从 i 到 j 的所有路径。如果删除那些点后的结果图不再会有 i 或 j，那么在 i 和 j 之间的所有路径就都被破坏了。

从图13—1 的线性图为例，点组（a，d，f）不是一个分开 i 和 j 的最小点组，也就不是一个分开 i 和 j 的点截割，这是因为只要删除点 a 和 d 就将破坏 i 和 j 之间的所有路径。可是只删除点 a 或 d（但不是 a 和 d 两者）不能破坏 i 和 j 之间的所有路径。所以点组（a，d）是分开 i 和 j 的一个点截割，删除（a，d）的结果见图13—4。在图13—1 中分开 i 和 j 的所有点截割是（i）和（j），（a,b,f）和（a,d）。

为了研究一个点权交通网络中的最大流量，我们先要对一个饱和点截割下定义。

当分配给一个点的一个流量等于这个点的点负载量时，这个点就饱和了。如果在一个点截割的点组中的所有点都饱和了，这个点截割就叫做饱和点截割。

作为一个饱和点的例子，我们可以观察在图13—2中所分配的一个流量。因为分配给点a的一个流量为1，而点a的点负载量为1，所以点a是饱和了。注意，如果一个点饱和了，我们就不能再分配任何流经这个点的流量了。在这同一图中，点截割（a，d）和（a，b，f）是饱和的，可是，我们可以看出在一个指定流量的情况下存在着饱和点截割并不表明这个网络有了最大流量了。在另一方面，如果有一个分开i和j的饱和点截割，我们就不能再从i到j分配另外流量了。

定理13—1：当分配一个从i到j的流量ψ_{ij}时，如果有一个分开i和j的饱和点截割的话，那么从i到j就不能再分配另外（非零）流量。

证：因为任何从i到j的路径将经过分开i和j的点截割中的一个点，并且因为在点截割中所有点是饱和的。这就明显地不能再分配从i到j的另外流量了。▌

与边权交通网络中的基础饱和截割组相似，在点权交通网络中，我们对基础饱和点截割下定义如下：

一个分开i和j的饱和点截割在分配一个流量时点截割是饱和了。如果从i到j的每一条路径当分配一个非零流量时，确实只有一个点在这个点截割的话，那么这个饱和点截割就叫做基础饱和点截割。

例如，在图13—2中，分配流量如虚线所示。组（a，d）虽然是一个饱和点截割，但却不是一个基础饱和点截割。这是因为当一个非零流量$\psi_{1_{ij}}$分配给一条路径（e_1，e_4，e_5，e_8）时，这个流量通过两个点a和d的。在图13—3的流量分配中，因为$\psi_{1_{ij}}$和$\psi_{3_{ij}}$只通过组（a，d）中的一个点，所以（a，d）是一个基础饱和点截割。同样，在图13—3中，组（a，b，f）也是一个基础饱和点截割。有了这个定义，我们有下面一个重要定理。

定理13—2：对于一个无向点权交通网络，当且仅当分开i和j有一个基础饱和点截割时，从i到j的流量是最大值。

在证明这个定理以前，让我们先学习一个辅助定理。

辅助定理13—1：设W_1和W_2是分开点i和点j的两个点截割。设g_2是含有点j的一个部分g_1是删除W_1中所有点后所得到余留下来的子图。注意g_1和g_2可能只包括一些点。如果W_2含有g_1和g_2两者中共有的一些点时，那么就有一个分开i和j的点截割W，这样使$W \subset W_1 \cup W_2$，并且W不含有g_1中任何点。

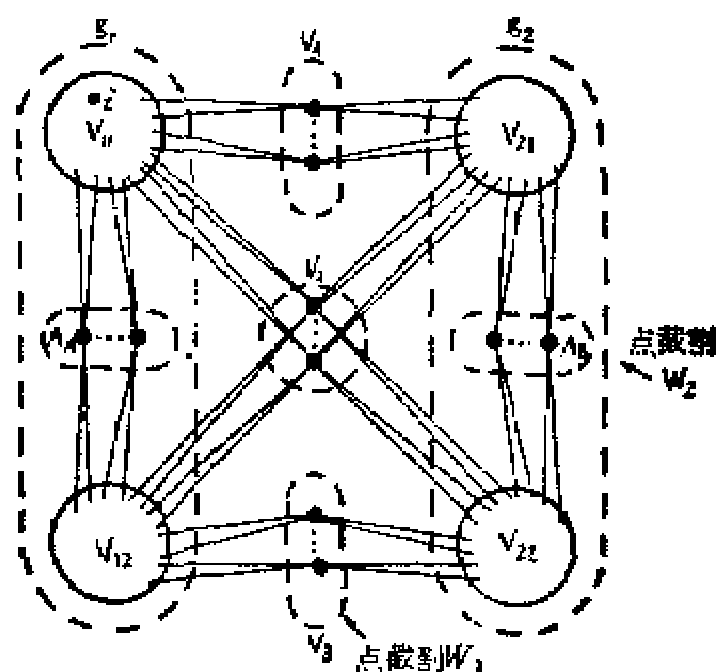


图 13—5

证：在图13—5中的线性图中，图中所有各点是在点不共用组V_{11}，V_{12}，V_{21}，V_{22}，V_A，V_I，V_B，A_A和A_B中，点截割$W_1=V_A \cup V_I \cup V_B$，点截割$W_2=A_A \cup V_I \cup A_B$，和$i \in V_{11}$。注意$g_1$包括$V_{11}$，$V_{12}$和$A_A$中所有各点以及连接这些点之间的所有边。也要注意到在图13—5中的线性图是够一般性的。足以用来证明这个辅助定理的。

因为$j \in g_2$，j可以在V_{21}中，也可以在V_{22}中，当$j \in V_{21}$时，组$V_A \cup A_B \cup V_I$清楚地含

有分开i和j所要求的点截割。同样，当$j\in V_{22}$时组$A_B\cup V_i\cup V_B$含有分开i和j所要求的点截割。▌

设W_0是分开i和j的一个饱和点截割，g_2是含有点j的一个部分，和g_1是删除W_0中所有点后所得到的余留下来的子图。如果没有其它分开i和j的饱和点截割含有g_2中一些点，那么W_0就叫做最接近点j的饱和点截割。

在图13—2中，因为没有其它分开i和j的饱和点截割含有点j，点截割（a，d）是最接近点j的饱和点截割。注意在这种情况的g_2包括一个点，那就是j。再注意这样一个点截割的存在可由辅助定理13—1所保证。

有了最接近某点的饱和点截割的定义，我们可以证明定理13—2如下：如果没有分开i和j的饱和点截割，我们就可以从i到j再分配另外流量。因为如果对一个无向点权交通网络G已经分配过从i到j的最大流量，我们可以假定分开i和j的至少有一个饱和点截割。

假使分开i和j的有k个饱和点截割，但是没有一个是基础饱和点截割。设W_0是最接近点j的饱和点截割如图13—6所示。因为W_0不是一个基础饱和点截割，就有一条从i到j的路径，它通过W_0中不止一个点，对这点路径已经分配过一个非零流量ψ_{1ij}设这条路径为P，它通过W_0中的第一个点是v_1如图13—6所示，那么从v_1到j只通过g_2中的一些点的有一条路径P_1，对这条路径可以分配一个从v_1到j的非零流量。设从i到j由路径P_1和从i到v_1的路径P中一部分所组成的路径为P'_0把分配过给P的流量改变为$\psi_{1ij}-\delta$而把某些非零δ分配给P'，那么我们仍能保持从i到j的同样流量。可是因为在W_0中至少有一个点的流量减少了δ，所以W_0不再是饱和的了。因此改变从i到j的流量分配后，饱和点截割数就减少了，并且只能产生基础饱和点截割。在分开i和j的点截割中至少有一个饱和点截割，但是没有基础饱和点截割的情况下，这是可能的。因此我们可以把饱和点截割数k减少到零。因为当没有分开i和j的饱和截割时，我们可以分配另外流量。所以分配的流量不能是最大量。这是一个矛盾。因此至少有一个分开i和j的基础饱和截割。

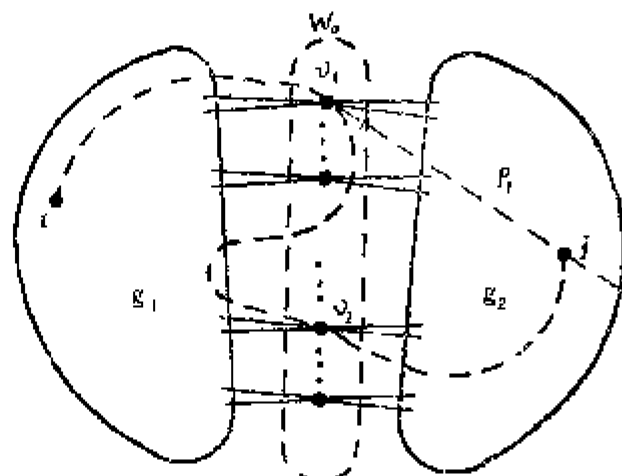


图 13—6

定理13—2的逆定理也是正确的。即对于一个无向点权交通网络，如果从i到j的流量是最大量，那么至少有一个分开i和j的基础饱和点截割。

我们用符号$\cup(W_i)$表示在组W_i中所有点的点负载量之和，这叫做组W_i的值。如图13—1中点组（a，d）的值$\cup(a,d)=1+2=3$。

当从i到j分配有最大流量时，根据定理13—2就有一个分开i和j的基础饱和点截割。每一个从i到j分配有一个非零流量以得到总的最大流量的路径，根据定义，只通过一个基础饱和点截割中的一个点。所以，如果W_0是分开i和j的一个基础饱和点截割，那么W_0值一定等于从i到j的最大流量。正象在边权交通网络中的情况那样，如果我们用一个极限负载量t_{ij}来表示从i到j的最大流量，我们有

$$t_{ij}=\cup(W_0) \qquad (13—2)$$

设$\{W\}$是分开i和j的所有可能的点截割，那么我们可以知道对于在$\{W\}$中每一个W，

$$\cup(W_0)\leqslant\cup(W) \qquad (13—3)$$

定理13—3：对于一个无向点权交通网络，从 i 到 j 的极限负载量 t_{ij} 等于

$$t_{ij}=\min[\bigcup(W)\mid W\in\{W\}] \tag{13—4}$$

注意这个定理与定理12—3之间的相似性。

例13—1　在图13—1中的无向点权交通网络 G 中，分开 i 和 j 的所有可能点的截割（i），（j），（a，b，f）和（a，d），这些点截割的值是

$$\bigcup(i)=4$$
$$\bigcup(j)=5$$
$$\bigcup(a,b,f)=3$$
$$\bigcup(a,d)=3$$

因此从 i 到 j 的极限负载量 t_{ij} 是

$$t_{ij}=\min\{4,5,3,3\}=3$$

当考虑从 i 到 d 的最大流量时，我们分开 i 和 d 的所有可能的点截割，它们是（i），（d），（a，b，f）和（b，f，j）。所以从 i 到 d 的极限负载量为

$$t_{id}=\min\{4,2,3,7\}=2$$

第二节　有向点权交通网络

当一个点权交通网络的每一条边都是定向的，这个网络叫做有向点权交通网络。例如图13—7中的网络是一个有向点权交通网络，每条边所指的定向是这条边所能输送的流量方向。所以为了分配一个从 i 到 j 的流量，我们必须分配一个从 i 到 j 沿着一个有向路径的流量。例如我们可以对图13—7中的网络如下分配流量 $\psi_{ij}=3$。对一条有向路径 $P_1=(e_1,e_2)$，我们分配流量 $\psi_{1ij}=1$。因为

$$1\leqslant\min\{c_i,c_a,c_j\}$$

所以这是允许的。再对一条有向路径 $P_2=(e_4,e_5,e_8)$ 分配流量 $\psi_{2ij}=1$。因为

$$1\leqslant\min\{c_i-\psi_0(i),c_f-\psi_0(f),c_d-\psi_0(d),c_j-\psi_0(j)\}$$

其中 $\psi_0(i)=1$，$\psi_0(f)=0$，$\psi_0(d)=0$，$\psi_0(j)=1$，$c_i=4$，$c_f=1$，$c_d=2$，$c_j=5$，这也是可行的，最后对一条有向路径 $P_3=(e_6,e_7,e_8)$ 分配流量 $\psi_{3ij}=1$，如图13—8所示。

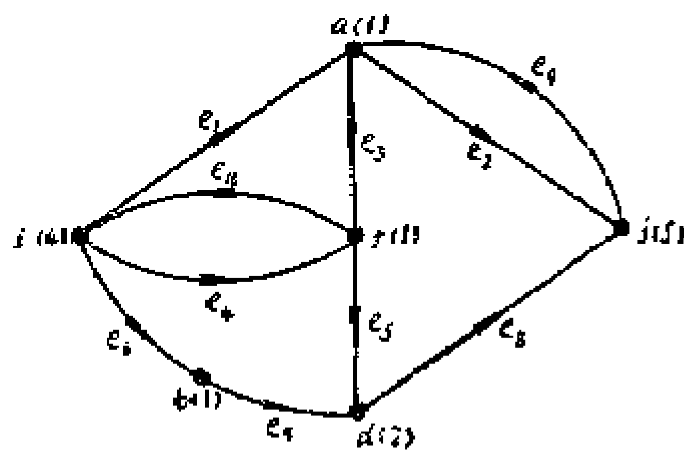


图　13—7

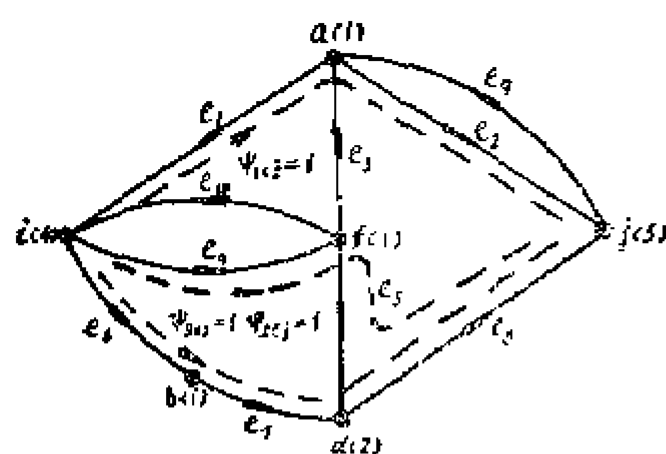


图　13—8

对于有向点权交通网络，从拓扑学的角度来求取最大流量要应用到点半截割而不是点截割。

所谓从 i 到 j 的点半截割是一个最小的点组，删除了这个组中的所有点破坏了从 i 到 j

的所有有向路径。如果删除那些点后的结果图不含有点 i 或点 j，这就是破坏了从 i 到 j 的所有有向路径。

例如，图13—7中的有向点权交通网络中的点组（a，d）是一个从 i 到 j 的点半截割，也是一个分开i和j的点截割。设$S=E(V_1\times\overline{V_1})\cup E(\overline{V_1}\times V_1)$是分开$i$和$j$的截割组，其中$i\in V_1$，那么$S$可以分解成两个半截割$s_{ij}$和$s_{ji}$

$$s_{ij}=E(V_1\times\overline{V_1}) \tag{13—5}$$

$$s_{ji}=E(\overline{V_1}\times V_1) \tag{13—6}$$

它们有下列特性：

$$s_{ij}\cap s_{ji}=\phi \tag{13—7}$$

可是点半截割可能没有这样一个特性。例如图13—7中的有向点权交通网络中的点组（a，d）是分开 i 和 j 的一个点截割。子组（a）是从 j 到 i 的一个点半截割，可是子组（d）却不是从 i 到 j 的一个点半截割。从 i 到 j 的一个点半截割，即组（a，d）的一个子组是（a，d）本身。因此这两个点半截割有一个共同点。

有了点半截割的定义后，我们有下面一条与定理12—5相类似的定理。

定理13—4：在一个有向点权交通网络中，从 i 到 j 的一个极限负载量t_{ij}等于

$$t_{ij}=\min\{\cup(w_{ij})\} \tag{13—8}$$

其中$\{U(w_{ij})\}$是从i到j所有点半截割值的集体。注意t_{ij}等于从i到j的最大流量，而点半截割值是点半截割中所有点的点负载量之和。

例13—2　在图13—9中有一个有向点权交通网络。为了求取从i到j的极限负载量t_{ij}，首先我们确定从i到j的所有点半截割：（i），（a,c），（b,c），（b,d）和（j）。这些半截割的值是

$\cup(i)=4$

$\cup(a,c)=4$

$\cup(b,c)=5$

$\cup(b,d)=3$

$\cup(j)=5$

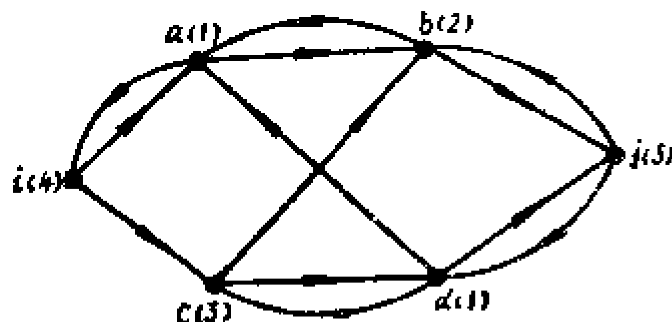


图　13—9

所以根据定理13—4极限负载量t_{ij}为

$$t_{ij}=\min\{4,4,5,3,5\}=3$$

为了求取从j到i的极限负载量t_{ji}，我们先确定从j到i的所有点半截割为{(j)，(b,d)，(a)，(i)}那么t_{ji}为

$$t_{ji}=\min\{\cup(j),\cup(b,d,)\cup(a),\cup(i)\}=\min\{5,3,1,4\}=1$$

第三节　点截割和点半截割的产生

在上面几节中我们已经看到分开 i 和 j 的点截割和从 i 到 j 的点半截割是点权交通网络中的重要点组，尤其当我们需要求取从 i 到 j 的极限负载时，对于知道无向点权交通网络中分开 i 和 j 的所有点截割和有向点权交通网络中从 i 到 j 的所有点半截割是有必要的。所以为了求取点权交通网络的极限负载量，我们要研究产生点截割和点半截割的方法。首先让我们复习一下如何产生分开 i 和 j 的所有截割组的。

从第四章的第五节中我们已经学过所有截割组，截割组的边不共用并集，以及空组，通

过环和运算是一个阿贝耳群。所以我们可以用已知的生成元来产生所有截割组。而且，如果我们把截割组和截割组的边不共用并集分为两组，即$\{S(i,j)\}$和$\{S(ij,\ )\}$，其中$\{S(i,j)\}$包括（1）分开i和j的所有截割组，和（2）含有分开i和j的奇数截割组的一些截割组的边不共用并集，而$\{S(ij,\ )\}$包括不属于$\{S(i,j)\}$的所有截割组和截割组的边不共用并集，那么我们有

（一）如果S_1和S_2在$\{S(i,j)\}$之中，那么$S_1\oplus S_2$是在$\{S(ij,\ \}$之中。

（二）如果S'_1和S'_2在$\{S(ij,\ )\}$之中，那么$S'_1\oplus S'_2$是在$\{S(ij,\ )$之中。

（三）如果S_1在$\{S(i,j)\}$之中而S_2在$\{S(ij,)\}$之中，那么$S_1\oplus S_2$是在$\{S(i,j)\}$之中。

有了这些特性，如果我们有一组生成元，而且知道哪些生成元是分开i和j的，我们就能从这些生成元中产生分开i和j的截割组。特别是如果我们选择一组生成元S_0，$S_1,\cdots$，S_m（$m=n_e-n_v+k$），其中只有S_0分开i和j的那么我们知道所有分开i和j的截割组是在集体

$$\{S(i,j)\}=\{S_0\oplus S_K|S_K\in\{S'\}\} \tag{13-9}$$

之中，其中$\{S'\}$包括空组，截割组S_1，$S_2\cdots$，S_m以及由S_1，S_2，$\cdots$，S_n的所有可能组合的环和所得到的那些组。而且如果在集体$\{S_0\oplus S_K|S_K\in\{S'\}\}$之中的$S_a$不是一个截割组，那么在这同一集体中另有一个$S_b$，$S_a\supset S_b$（即$S_b\subset S_a$）。因此在边权交通网络情况下，极限负载量等于

$$t_{ij}=\min\{U(S_a\oplus S_k)|S_k\in\{S'\}\} \tag{13-10}$$

不幸的是分开i和j的点截割集体没有这些好的特性。所以产生分开i和j的所有点截割不象分开i和j的所有截割组那样简单。可是，由于下面几条定理所给出的一些特性，我们可以用产生分开i和j的截割组的方法来得到分开i和j的所有点截割。

定理13—5：对于一个点权交通网络G，设W是分开i和j的一个点截割，和g_1和g_2是从G中删除W中所有点后所得到的两个部分。设除当$W=(j)$外，点j是在g_2之中，那么g_2是一个零图（它不含点）。因此如果$V(g_2)\neq\phi$，那么$E[W\times V(g_2)]$是分开i和j的一个截割组，其中$V(g_2)$是g_2中的所有点的点组。

证：从点截割的定义可知，从i到j的每条路径一定通过点截割W中至少一个点的。因为$E[W\times V(g_2)]$中的每一条边是关联到W中的一个点，从i到j的每一条路径一定含有$E[W\times V(g_2)]$中至少一条边。所以从一个网络中删除$E[W\times V(g_2)]$中所有边将破坏从i到j的所有路径。因此$E[W\times V(g_2)]$含有从i到j的一个截割组。设g'_1和g'_2是从图中删除$E[W\times V(g_2)]$中所有边后所得到的两个子图，并且g'_2是一个含有点j的部分。可以看出g'_2和g_2是一样的，而且$V(g'_1)=V(g_1)\cup W$，所以g'_1包括$V(g_1)\cup W$中两点之间所连接的那些边。现在g'_1一定是连通的，否则就会在W中至少有一个点使所有从i到j的路径都不通过这个点。这是与W是一个分开i和j的点截割的假设相矛盾的。因为$E[W\times V(g_2)]$是一个连接g'_1和g'_2的边组，所有$E[W\times V(g_2)]$是一个最小边组，删除了这个边组中所有边后使得一个连通图分开了。因此$E[W\times V(g_2)]$是分开i和j的一个截割组。∎

例如，在图13—10（a）中有一个点权交通网络。对于一个分开i和j的点截割（a，b）。删除这个点截割（a，b）后所得到的两个部分g_1和g_2如图13—10（b）所示。所以$V(g_2)=(c,d,j)$，$E[W\times V(g_2)]=(e_4,\ e_6,\ e_8)$是分开$i$和$j$的一个截割组。对于一个点截割（$a$，$d$）而言，$V(g_2)=(a,b,c,d,i)$，和$E[W\times V(g_2)]=(e_1,\ e_2)$，这

也是一个分开 i 和 j 的截割组。

根据定理13—5除点截割$W=(j)$外，我们可以从每个W得到一个截割组$E[W\times V(g_2)]$。为了方便起见，我们称截割组$E[W\times V(g_2)]$为点截割W的对应截割组，其中g_2是删除了W中所有点后所得到的一个含有点j的部分。

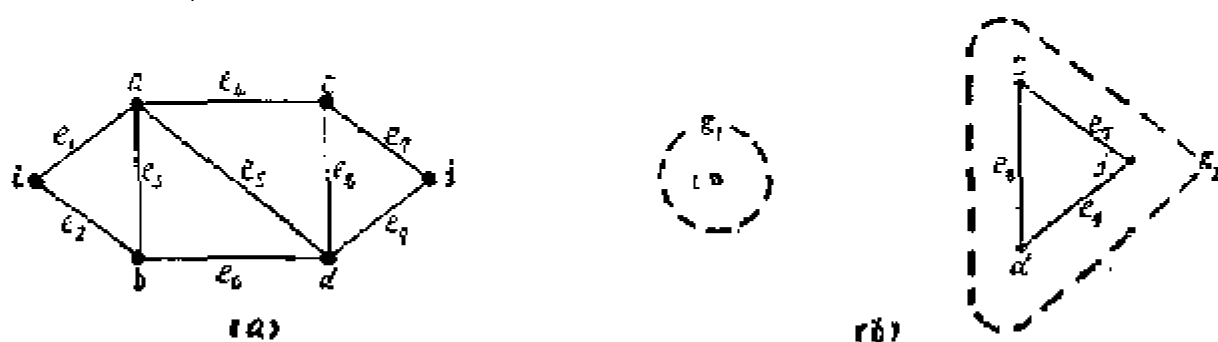


图 13—10

我们想知道是否任何一个分开 i 和 j 的截割组S就有一个分开 i 和 j 的点截割，而S成为这个点截割的对应截割组。答案是"否"。作为一个例子，在图13—5（a）中$S=(e_2, e_3, e_4, e_6)$是分开 i 和 j 的一个截割组，但是没有一个分开 i 和 j 的点截割，它的对应截割组是S。

定理13—**6**：假设S是一个连通图G中分开 i 和 j 的一个截割组。设g_1和g_2是S的对应子图（就是g_1和g_2是从G中删除S中所有边后所得到的部分），假设$i\in g_1$那么

$$W=V(S)\cap(g_1) \tag{13—11}$$

所得出的点组W含有一个分开 i 和 j 的点截割，其中$V(S)$是S中所有边的端点组。

证：这里只需证明删除W中所有点后将破坏从 i 到 j 的所有路径。因为S中每条边是连接着g_1中的一个点与g_2中的一个点的，所以式（13—11）中所给出的组W有着S中每条边的两个端点之一的。所以删除W中所有点将删除S中所有边的。我们知道删除S中所有边破坏从 i 到 j 的所有路径，因此删除W中所有点破坏从 i 到 j 的所有路径。∎

例如，在图13—10（a）中，$S=(e_5, e_6, e_7, e_3)$是分开 i 和 j 的一个截割组。现在$V(S)$是（a,b,c,d,j）和$V(g_1)$是（i,a,b,c）。所以由式（13—11）所给出的组W是（a,b,c），很清楚这个组含有分开 i 和 j 的点截割（a，b）。

为了方便，在定理13—6中的组W叫做截割组S的对应点组，用符号$W(S)$表示。例如，在图13—10(a)中点权交通网络的一些截割组S_1，S_2，S_3和S_4，其中$S_1=(e_1, e_2)$，$S_2=(e_1, e_3, e_6)$，$S_3=(e_2, e_3, e_4, e_5)$，$S_4=(e_4, e_5, e_6)$它们的对应点组是：

$$W(S_1)=(i)$$
$$W(S_2)=(i,b)$$
$$W(S_3)=(i,a)$$
$$W(S_4)=(a,b)$$

假设在图13—10中的$\{S(i,j)\}$中的S不是一个截割组而是一些截割组的边不共用并集，那么我们可以如下组成对应S的一个点组W。设g_1和g_2是从G中删除S中所有边后所得到的两个边不共用的子图，而g_1是含有点 i 的一个部分，那么组W是在S中也是在g_1中的所有边的端点的集体。

注意删除S中所有边后产生$k(>2)$个部分，而其中之一含有点 i 的一定要选作g_1。例如在图13—10(c)的$\{S(i,j)\}$中一些截割组S的一个边不共用并集是$S=(e_1, e_2, e_4, e_5, e_7, e_9)$，那么子图$g_1$就如图13—11所示，所以点组$W$是($i,b,c,d$)。

设S是$\{S(i,j)\}$中的一个组。设g_1是删除S中所有边后所得到的含有点i的一个部分。那么组S的对应点组是

$$W(S)=V(S)\cap V(g_1) \qquad (13—12)$$

其中$V(S)$是S中所有边的端点的点组，和$V(g_1)$是G中所有点的点组。

图 13—11

有了上述点组$W(S)$的定义，我们可以把定理 13—6 扩展为：

定理13—7：设S是$\{S(i,j)\}$中的一个组，那么组S的对应组$W(S)$含有分开i和j的一个点截割。

因为$\{S(i,j)\}$是容易产生的，如果我们能够把$\{S(i,j)\}$与一个点权交通网络的一个极限负载量t_{ij}联系起来，我们就会不费时间地得到t_{ij}。下面的定理就会给出这样一种关系。

定理13—8：一个点权交通网络的一个极限负载量t_{ij}等于

$$t_{ij}=\min\{U(j),U\{W(S)\}|S\in(i,j)\}\} \qquad (13—13)$$

证，除$W=(j)$，对于分开i和j的任何点组W在$\{S(i,j)\}$中有一个S，根据定理 13—3，它是W的对应截割组。从上述点组的定义出发，我们知道由式（13—12）所给出的点截割$W(S)$，它的对应截割组是S。因此集体$\{U(j)$，$U\{W(S)\}|S\in\{S(i,j)\}\}$含有集体$\{U(W)|$所有分开$i$和$j$的点截割$W\}$。而且，对于$\{S(i,j)\}$中任何组$S'$，根据定理13—7，组$W(S')$含有一个点截割，譬如说$W$。所以

$$\min\{U(j),\ U\{W(S)\}|S\in\{S(i,j)\}\}$$
$$=\min\{U(W)|\text{分开 } i \text{ 和 } j \text{ 的所有点截割} W\} \qquad (13—14)$$

因为根据定理13—3，式（13—14）的右方等于从i到j的极限负载量t_{ij}，这个定理是正确的。▌

根据这个定理，我们可以从下一个例子看出由产生集体$\{S(i,j)\}$来得到一个点权交通网络的一个极限负载量。

例13—3 从图13—12中的点权交通网络，除对应着点j以外的所有关联组是

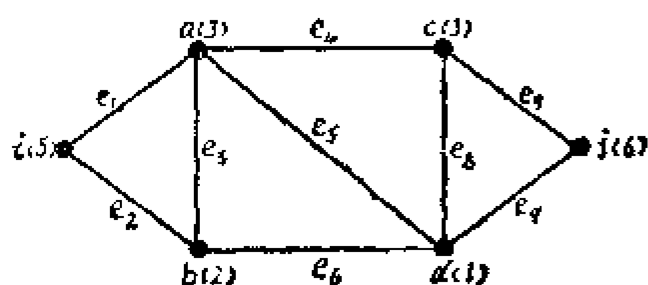


图 13—12

$$S(i)=(e_1,\ e_2)$$
$$S(a)=(e_1,\ e_3,\ e_4,\ e_5)$$
$$S(b)=(e_2,\ e_3,\ e_6)$$
$$S(c)=(e_4,\ e_7,\ e_8)$$
$$S(d)=(e_5,\ e_6,\ e_8,\ e_9)$$

因为只有$S(i)$是分开i和j的截割组，集体$\{S(i,j)\}$可由式（13—9）来表达。所以在$\{S(i,j)\}$，$W(S)$和$U\{W(S)\}$中所有组如表13—1所示：

$$t_{ij}=\min\{U(j),\ U\{W(S)\}|S\in\{S(i,j)\}\}=4$$

当一个点权交通网络是有向时，我们应该像前节所讨论的那样改用点半截割，为了得到点半截割，我们先要对对应点半截割下定义。

设G是一个线性图，和$V(G)$是G中所有点的点组，那么$V^+(G)$是$V(G)$的一个子组，其中G中每一条边是从$V^+(G)$中的一个点连接出来的。换句话说$V^+(G)$是G中所有点的点组，它的出次数$d^+(v)$不是零。

有了这个定义，$\{S(i,j)\}$中组S的对应点半截割可以定义如下。

$\{S(i,j)\}$，$W(S)$ 和 $U\{W(S)\}$　　表13－1

$\{S(i,j)\}$	$W(S)$	$U(j)$ 和 $U\{W(S)\}$
	(j)	6
$S(i)=(e_1,e_2)$	(i)	5
$S(i)\oplus S(a)=(e_2,e_3,e_4,e_5)$	(i,a)	8
$S(i)\oplus S(b)=(e_1,e_3,e_6)$	(i,b)	7
$S(i)\oplus S(c)=(e_1,e_2,e_4,e_7,e_8)$	(i)	5
$S(i)\oplus S(d)=(e_1,e_3,e_5,e_6,e_8,e_9)$	(i)	5
$S(i)\oplus S(a)\oplus S(b)=(e_4,e_5,e_3)$	(a,b)	5
$S(i)\oplus S(a)\oplus S(c)=(e_2,e_3,e_6,e_7,e_8)$	(i,a,c)	11
$S(i)\oplus S(a)\oplus S(d)=(e_2,e_3,e_5,e_6,e_8,e_9)$	(i,a,d)	9
$S(i)\oplus S(b)\oplus S(c)=(e_1,e_5,e_4,e_6,e_7,e_8)$	(i,b)	7
$S(i)\oplus S(b)\oplus S(d)=(e_1,e_2,e_6,e_3,e_9)$	(i,b,d)	8
$S(i)\oplus S(c)\oplus S(d)=(e_1,e_2,e_4,e_5,e_6,e_7,e_9)$	(i)	5
$S(i)\oplus S(c)\oplus S(b)\oplus S(c)=(e_5,e_4,e_7,e_8)$	(a,b,c)	8
$S(i)\oplus S(a)\oplus S(b)\oplus S(d)=(e_4,e_3,e_9)$	(a,d)	4
$S(i)\oplus S(a)\oplus S(c)\oplus S(d)=(e_2,e_3,e_6,e_7,e_9)$	(i,a,c,d)	12
$S(i)\oplus S(b)\oplus S(c)\oplus S(d)=(e_1,e_2,e_4,e_5,e_7,e_9)$	(i,b,c,d)	11
$S(i)\oplus S(a)\oplus S(b)\oplus S(c)\oplus S(d)=(e_7,e_8)$	(c,d)	4

设 S 是一个有向点权交通网络 G 的 $\{S(i,j)\}$ 中的一个组。设g是从 G 中删除 S 中所有边后所得到含有点 i 的一个部分。那么组 S 的对应点半截割是一个点组，用符号 $w(S)$ 来表示，由下式给出

$$w(S)=V^+(S)\cap V(g) \tag{13—15}$$

下面我们可以看到对于从 i 到 j 的任何一个点半截割，有一个分开 i 和 j 的截割组 S，使 $w(S)$ 是从 i 到 j 的点半截割。

定理13—9：在一个有向点权交通网络中，除了 $w=(j)$ 外，对于从 i 到 j 的任何一个点半截割 w，有一个分开 i 和 j 的截割组 S，使 w 有像式（13—15）所给出的那种关系。

证：我们取满足下列两条件的所有一些边组成为一个边组 S_{ij}：

（一）每条边是从 w 中一个点连接出来的，它的定向是从 w 中的一个点出来的。

（二）每条边至少在一条从 i 到 j 的有向路径中，它正好通过 w 中的一个点。

很明显 S_{ij} 是从 i 到 j 的一个半截割。我们知道从 j 到 i 另有一个半截割 S_{ji}，$S_{ij}\cap S_{ji}=\phi$ 和 $S_{ij}\cup S_{ji}=S$，后者是分开 i 和 j 的一个截割组。有了这个截割组 S，我们可以得到一个含有点 i 的部分g。根据定理13—6，点组 $V(S)\cap V(g)$ 中所有点破坏了从 i 到 j 的所有有向路径和从 j 到 i 的所有有向路径。如要只破坏从 i 到 j 的所有有向路径但不破坏从 j 到 i 的所有有向路径，我们只需用 $V^+(S)$ 来代替 $V(S)$。换句话说，删除 $V^+(S)\cap V(g)$ 中的所有点将破坏从 i 到 j 的所有有向路径。因为 $S=S_{ij}\cup S_{ji}$

$$V^+(S)\cap V(g)=V^+(S_{ij})\cap V(g) \tag{13—16}$$

而且，我们知道

$$V^+(S_{ij})\cap V(g)=w \tag{13—17}$$

所以这个定理是正确的。▮

我们以图13—13中的有向点权交通网络为例，考虑一个点半截割（a，b）。图为半截

割$S_{ij}=(e_4, e_5, e_6)$，半截割$S_{ij}=(e_5', e_{10}', e_{11}')$，所以$S=(e_4, e_5, e_6, e_5', e_{10}', e_{11}')$。子图$G_1$包括点$i$，$a$和$b$，以及边$e_1$，$e_1'$，$e_2$，$e_2'$和$e_3$。点组$V(S)=(a, b, c, d, i, j)$和

$$V(S)\cap V(g)=(i, a, b)$$

因为$V^+(S)=(a, b, d, j)$所以

$$V^+(S)\cap V(g)=(a, b)$$

这就是给出的点半截割。注意这也等于

$$V^+(S_{ij})\cap V(g)。$$

与定理13—6相似，我们有定理13—10。

定理13—10：设S是有向点权交通网络G中$\{S(i,j)\}$中的一个组。设g是删除S中所有边后所得到的含有点i的一个部分。那么$V^+(S)\cap V(g)$含有从i到j的一个点半截割。

有了定理13—9和13—10之后，我们可以得到另外一个重要定理。

定理13—11：一个有向点权交通网络的从i到j的一个极限负载量等于

$$t_{ij}=\min\{U(j),\ U\{w(S)\}|S\in\{S(i,j)\}\} \qquad (13—18)$$

式中$W(S)$由式（13—15）给出。

这个定理的证明差不多与定理13—8的完全相同。根据这个定理，我们可以靠得到$\{S(i,j)\}$来求取一个有向点权交通网络的一个极限负载量。下面就是一个例子。

例13—3　在图13—13中的有向点权交通网络中，除对应点j以外的所有关联组为

$S(i)=(e_1, e_1', e_2, e_2', e_{11}')$

$S(a)=(e_1, e_1', e_3, e_4, e_5, e_5')$

$S(b)=(e_2, e_2', e_3, e_6, e_{10}')$

$S(c)=(e_4, e_7, e_8)$

$S(d)=(e_5, e_5', e_6, e_8, e_9, e_9', e_{11}')$

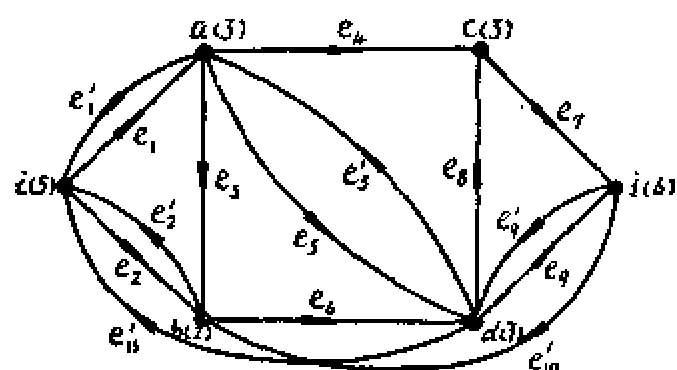


图　13—13

因为只有$S(i)$分开i和j，集体$\{S(i,j)\}$可由式（13—9）来表示。所以在$\{S(i,j)\}$，$w(s)$和$U\{W(S)\}$中所有组如表13—2所示。由这个表可知，$t_{ij}=4$。

第四节　关键路径法

在实际生活中很多规划，诸如房屋的建筑，会议的准备，新铁路信号设备的设计，或者一项科学工作的准备，都可以划分为若干单独成分，这些成分我们可以称之为任务。常常，这些任务之间是有约束的：譬如说某一任务没有完成之前，另一任务是不能开始的。另一方面，某些任务相互之间是没有联系的，例如在铁路信号施工中，挖电缆沟与室内安装设备之间是没有直接联系的。

因此我们一定能够对那些任务的顺序构成一个有向图，并且把这种分析推进到所需的时间问题，试图将任务的工作时间安排得最短。这里我们假定，只要我们愿意，我们可以同时做很多任务，而这些任务是互不相依的。

在任何一个实际的规划或工程中会出现有些任务只要短时间内完成，可以安排在计划期中任何一段中而不会影响其它任务的完成。可是有些任务是关键的，如果它们不能尽早完

成，或者实际上比计划时间为长，那么整个工程的完成就将推迟。所以我们要识别出这些关键任务，使计划者特别注意使这些任务及时完成。或者必要的话，从非关键任务中转移一部分人力或设备去帮助那些关键任务，以免影响整个工程的完成。这种计划方法称为关键路径法，或简称CPM法。

$\left\{S(i,j)\right\}, w(S), U(j)$和$U\left\{W(S)\right\}$　　　　表13－2

$\left\{S(i,j)\right\}$	$W(S)$	$U(j)$和$U\left\{W(S)\right\}$
	(j)	6
$S(i)$	(i)	5
$S(i)\oplus S(a)=(e_2,e'_1,\ e_3,e_4,e_5,e'_5,\ e'_{11})$	(i,a)	8
$S(i)\oplus S(b)=(e_1,e'_1,\ e_3,e_6,e'_{10},\ e'_{11})$	(i,b)	7
$S(i)\oplus S(c)=(e_1,e'_1,e_2,e'_2,\ e_6,e_7,e_8,\ e'_{11})$	(i)	6
$S(i)\oplus S(d)=(e_1,e'_1,e_2,e'_2,\ e_6,e'_6,\ e_6,e_8,e_9,\ e'_{10})$	(i,d)	6
$S(i)\oplus S(a)\oplus S(b)=(e_2,e_3,e'_5,e_6,e'_{10},\ e'_{11})$	(a,b)	5
$S(i)\oplus S(a)\oplus S(c)=(e_1,e'_2,\ e_3,e_5,e'_5,\ e_7,e_8,e'_{11})$	(i,a,c)	11
$S(i)\oplus S(a)\oplus S(d)=(e_2,e'_2,\ e_3,e_4,e_5,e_5,e_8,e'_9)$	(i,a,d)	9
$S(i)\oplus S(b)\oplus S(c)=(e_1,e'_1,\ e_3,e_4,e_6,e_7,e'_8,e_{10},e'_{11})$	(i,b)	7
$S(i)\oplus S(b)\oplus S(d)=(e_1,e'_1,\ e_3,e_6,e'_6,\ e_8,e_9,e'_9,\ e'_{10})$	(i,d)	6
$S(i)\oplus S(c)\oplus S(d)=(e_1,e'_1,e_2,e'_2,\ e_4,e_6,e'_6,\ e_8,e_7,e_8,e'_8)$	(i,c,d)	9
$S(i)\oplus S(a)\oplus S(b)\oplus S(c)=(e_6,e'_6,\ e_3,e_7,e_8,e'_{10},e'_{11})$	(a,b,c)	8
$S(i)\oplus S(a)\oplus S(b)\oplus S(d)=(e_4,e_8,e_9,e'_9,e'_{10})$	(a,d)	4
$S(i)\oplus S(a)\oplus S(c)\oplus S(d)=(e_2,e'_2,\ e_3,e_5,e_7,e_8,e'_9)$	(i,a,c,d)	12
$S(i)\oplus S(b)\oplus S(c)\oplus S(a)=(e_1,e'_1,\ e_3,e_4,e_6,e'_1,e_7,e_8,e'_9,\ e'_{10})$	(i,c,d)	9
$S(i)\oplus S(a)\oplus S(b)\oplus S(c)+S(d)=(e_7,e_8,e'_9,e'_{10})$	(c,d)	4

一个规划或工程可以用一个有向图来表示，其中点代表任务，有向边代表“x必须在y开始之前完成”的关系。每个点还与工作时间有关。

以铁路信号的车站电气集中联锁工程的施工为例，敷设电缆必须等待电缆沟挖好后才能敷设，电动转辙机必须等待道岔整治后才能安装。所以每一项任务有它的准备工作，同时每一任务又有完成这一任务的工作时间。

例如，对于一个车站的电气集中联锁施工工程，大约包括下列任务：

编号	任务	先行任务	工作时间（天）
1	勘测和准备		5
2	编制预算	1	20
3	作施工图	1	10
4	备料	2，3	10
5	挖电缆沟和过道管	3	30
6	安装轨道电路的准备	3	20
7	工务整治道岔	1	30
8	敷设电缆	4，5	10
9	安装分线盘	2	30
10	安装信号机及轨道电路	8	15
11	安装室内设备	2	14
12	室内配线	4，11	37
13	安装转辙机	7	15
14	室外配线等	6，10，13，	30
15	联锁试验	9，12，14.	7

现在我们可以画出这个施工工程的任务网络或网络有向图，如图13—14所示。在上面的列表中，任务网络各点的编号最好形成一个逻辑编号，以后计算时会方便些。

在网络有向图中，首先要辨明哪些任务的开始与其它任务的完成无关。把这种任务用一个点画在纸的左边。如果这种点不止一个，就在更左边标一个起点 α，连接到所有这种点。

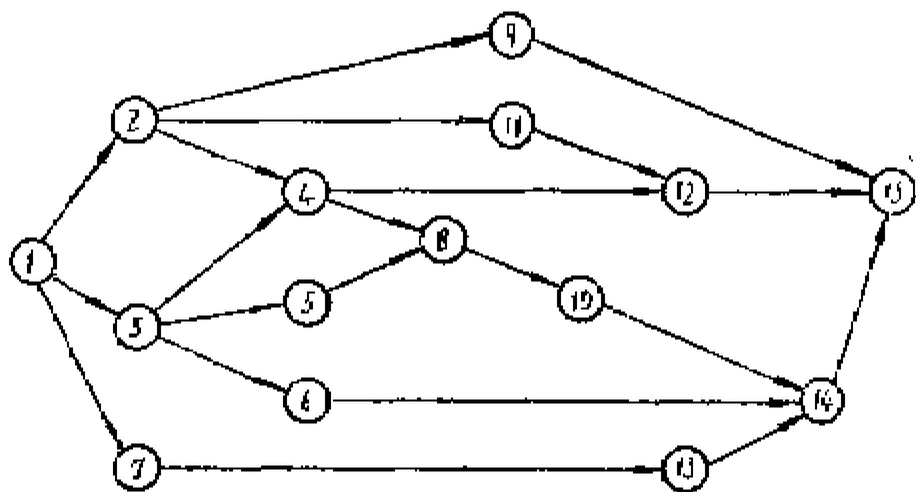


图 13—14

现在要辨明哪些任务的开始必须等待纸上已画出那些点的完成。把前者用有向边连接到已画点的左方，有向边的方向指明任务的依赖关系。可能有些点是相互依赖的，就应用相应的有向边来表明这种关系。注意，有向图中不应有有向回路。如果发现有有向回路，这就表明工程订得不好，某几个任务的前后关系必须修改，使得不能形成有向回路，然后再继续画下去。最后，所有任务都已画上。可能，至少有一个任务，别的任务不必等待它的完成。如果这种点只有一个，这就是终点，表示网络已经完成。如果这种点不止一个，那么把这些点用有向边连接到一个更右边的新点 ω，这个点就是终点。上列各个任务，如果照这个原则画出，即成图13—14。一个任务网络是一个连通的加权的、简单的、非循环的、并且只有一个起点和一个终点的有向图。

到现在为止，我们还没有谈论到每一任务的实际完成时间。我们假定对于每一个任务的工作时间是一个给定的常数。但是在很多实际生活中这是不可能的，时间常会是一个随机变数。在这种情况时，时间因素要估计到，因此常常采用计划评审技术来分析网络。可是增加了复杂性，因此这里仍然假定工作时间为常数。

为了一个规划构筑一个网络图的困难是有两方面的。第一是找整个规划的最短的可能的

工作时间，第二是设计一个时间进度表，使规划在这段时间内完成的机会最大。为了达到这个目的，我们在任务网络的相应任务点上标以每一个任务的工作时间。各任务之间必须遵照前后关系，整个规划的最短可能时间就对应着网络中从头到尾的最长路径。这个路径的长度是沿途各点的工作时间之和。这条路径或几条路径称为关键路径，在这条路径上的那些任务称为关键任务。如果任何一个关键任务耽误了时间，整个规划的完成就将延迟。不是关键的任务称为非关键的，它们的工作时间即使耽误某一段时间是不会影响整个规划的完成的。因此对于每个任务，我们可以发现它的可能开始的最早动工时间ESi是依赖于其它任务的完成。我们规定起点任务的最早动工时间作为时间零。我们也可以发现如果规划是在最短的可能时间内及时完成，那么每一任务的最晚完工时间LFi是可以完成的。

很明显，一个关键任务的最早动工时间ESi等于它的最晚动工时间LSi，就是说要求一个工程及时完成，时间方面是没有伸缩余地的。可是，对于非关键任务，最晚动工时间LS_i可比最早动工时间ES_i为迟。在这种情况时，这种伸缩余地称为宽裕时间。有两种宽裕时间将在下面说明。

所以一个任务网络的一个关键路径分析的结果是一个时间进度表，它表明每一个任务的最早和最晚动工时间以及最早和最晚完成时间。如果我们要求及时完成规划，我们必须着重考虑那些关键任务。进度表中也将显示出每一非关键任务的宽裕时间，这些宽裕时间可以用来弥补关键任务的不足，以免耽误规划。

如果非关键任务的延迟时间把所有宽裕时间都用完了，那么非关键任务也就成为关键的了。所以我们定义一个任务的最晚动工时间有一个限度，就是不致耽误整个工程的完工时间。相反地，我们定义最早完工时间为一个任务依赖其它较早任务完成之后可能完成的最早完工时间。

现在我们来计算一个网络中每一任务的这些时间。

设ES_i＝任务 i 的最早动工时间，

LF_i＝任务 i 的最晚完工时间，

LS_i＝任务 i 的最晚动工时间，

EF_i＝任务 i 的最早完工时间，

Ti＝任务 i 的工作时间。

如果α和ω是虚设任务T_α和T_ω，工作时间为零。各点的编号合乎逻辑，则较方便。

我们把表示每一个点的圈分成如图13—15所示，在画任务网络时，各点的编号 N_i 和它的工作时间T_i是立即可以写上的。其它数字用下列算法填上。

计算任务时间算法

（一）最早时间的计算法

最早时间与下列方程有关：

1． $ES\alpha=0$，所有时间由源任务的开始时算起。

2、 $EFi=ESi+Ti$，这是因为每一任务需用时间Ti。

3． $ESi=\max\{EFj\}$系对网络的一条有向边 (j,i) 的所有点 j 取最大值。所有任务必须等待在它以前的任务完成之后才能开始进行。

因为网络是一个非循环有向图，在确定$ES\alpha$后，一定总有一个点，对于许多先行点j的所有EFj是已知的。如果标号 i 是一种逻辑编号，各点就可按编号次序来考虑。作为一个计算的例子，可以图13—14的一部分来考虑点 8 和它的先行点。假使对于每一个点的最早动

工时间已经算出，每一点的最早完工时间可以把工作时间加到最早动工时间来得到的。由图13—16可见最早完工时间中的最大者是100天（对点14而言），这就是任务 15 的最早动工时间。因为15的工作时间$T_{15}=7$天，所以15的最早完工时间为107天。整个工程的工作时间等于107天。

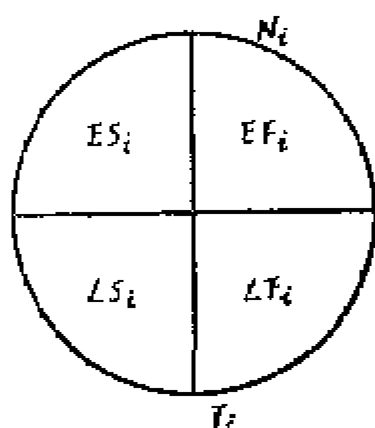


图 13—15

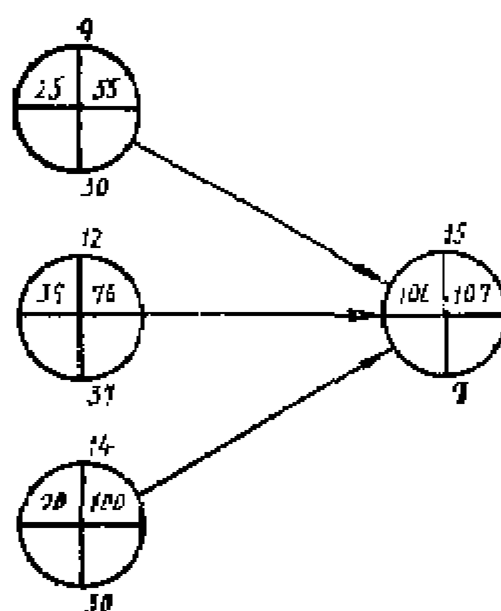


图 13—16

（二）最晚时间的计算法

知道了整个工程的最早完工时间EF_ω后，我们就可以计算为了在EF_ω内完成这个工程的各个任务的最晚动工和最晚完工时间。

反映最早时间的各方程为：

1． $LF_\omega=EF_\omega$

2． 因为每一任务 i 必须在Ti完工之前开始，所以$LSi=LFi-Ti$。

3． 对于 $i\neq\omega$的各任务，$LFi=\min\{LSi\}$，系对网络的一条有向边（i,j）所有点 j 中取最小值。

注意所有任务必须在任何后继任务最晚动工时间开始之前完工。

因为网络是非循环的，在规定LF_ω后，一定总有一个点，它对于许多后继点j 的LSj是已知的，如果标号 j 是一种逻辑编号，各点就可按编号的倒次序来考虑。

图13—17是图13—14的一部分，以此作为计算的一个例子。假设 4， 5， 6 的最晚完工时间是已知的，可以分别减去各项任务的工作时间来得到它们各自的最晚动工时间。点 3 的最晚动工时间是15天，那是最少的。因为任务 3 的工作时间为10天，所以它的最晚动工时间为 5 天。

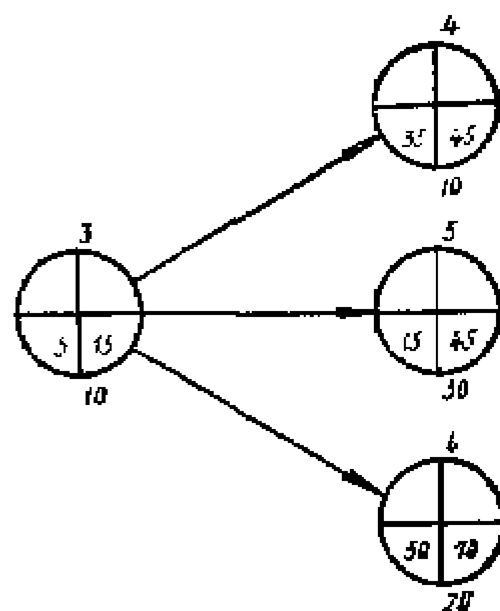


图 13—17

作为一个检查，α的最晚动工时间应为零，即$LS_\alpha=0$。

计算出图13—17右面的三个任务的四种时间后，现在我们可以判辨关键任务了。凡是任务 i 的$ESi=LSi$，这就是关键任务。这种任务也有$EF_i=LF_i$。从开始点到终点之间，几个关键任务形成一条或几条关键路径。在图13—18中所有任务的时间都已经计算出来。从这图中，可以看出：任务 1， 3， 5， 8， 10，14和15是关键的，形成一条关键路

径。整个工程的施工时间为107天。

决定了关键路径以后，下一工作是决定宽裕时间。一个任务 i 的总宽裕时间TFi是完成任务的最大容许时间（$LFi-ESi$）与它的工作时间Ti的差数就是$TFi=LFi-ESi-Ti=LSi-ESi=LFi-EFi$

在图13—18中，任务11的最晚完工时间是63，它的最早动工时间是25，所以任务11的最大容许工作时间是63－25＝38天。

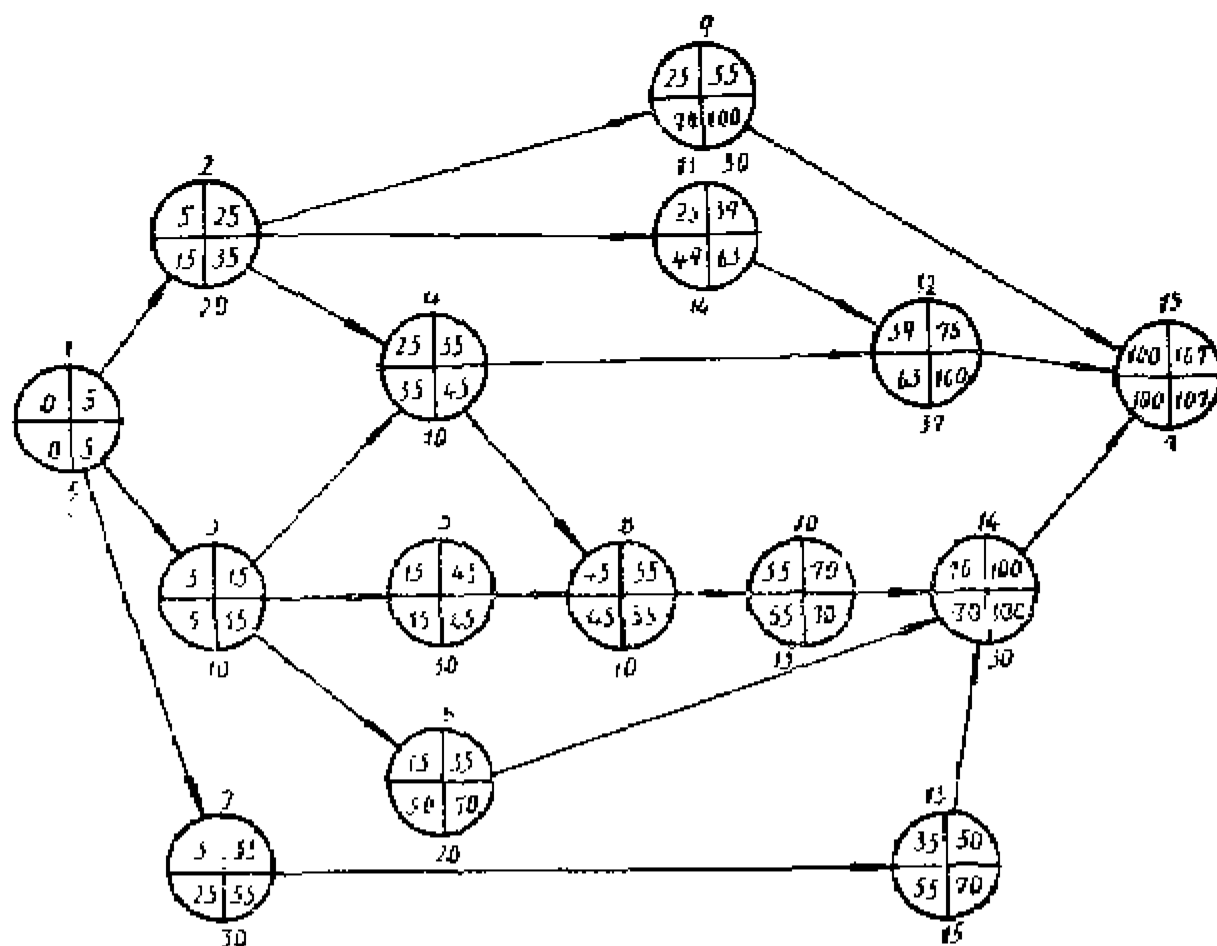


图 13—18

因为任务11只要求14天完成。所以对于该任务我们有一个总宽裕时间27天。当然，关键任务是没有总宽裕时间的。

第二种宽裕时间称为自由宽裕时间。任务 4 的最早完工时间为35天。任务 8 和12直接接在任务 4 之后。任务12有一最早动工时间39天，假定我们要尽快推动各任务，暂时可以忽略任务 8 ，那么任务 4 有一宽裕时间为39－35＝ 4 天。但是任务 8 也直接接在任务 4 之后，前者的最早动工时间为45，因此任务 4 有一宽裕时间为45－35＝10天。我们计算 4 的自由宽裕时间为两者中的较小者，即 4 天。一般，我们定义一个任务的自由宽裕时间FFi为

$$FFi=\min\{ESi-EFi\}$$

式中对网络中的所有有向边（i,j）取最小值，终点15没有其他任务接在后面，所以它的自由宽裕时间为零。如果列举工程中的各任务时最后出现两个终点，那么应该再在网络的右端加一个虚设任务作为真正的终点。如图13—19所示。当然，虚任务ω的工作时间为零，而它的FS_{ω}，LS_{ω}，LF_{ω}的时间都是前两个终点任务的 LF。

当一个任务没有总宽裕时间时，它也没有自由宽裕时间。例如关键任务就是这样。可是反过来说就不正确。一个任务可以没有自由宽裕时间，但是会有总宽裕时间的。任务 2， 7 和11就是这种例子。

现在我们用表格的方式来表达整个工程的时间进度如表 3 — 2 。

工程的时间进度表　　表 3—2

任　务	先行任务	T_i (天)	ES_i (天)	LS_i (天)	EF_i (天)	LF_i (天)	TF (天)	FF_i (天)
1		5	0	0	5	5	0	0
2	1	20	5	15	25	35	10	0
3	1	10	5	5	15	15	0	0
4	2,3	10	25	35	35	45	10	4
5	3	30	15	15	45	45	0	0
6	3	20	15	50	35	70	35	35
7	1	30	5	25	35	55	20	0
8	4,5	10	45	45	55	55	0	0
9	2	30	25	70	55	100	45	45
10	8	15	55	55	70	70	0	0
11	2	14	25	49	39	63	24	0
12	4,11	37	39	63	76	100	24	24
13	7	15	35	55	50	70	20	20
14	6,10,13	30	70	70	100	100	0	0
15	9,12,14	7	100	100	107	107	0	0

在计算出上表所列的六种时间后，计划人员就可以进行合理的管理。在这时间，如果任务的完成时间与指定的完成时间比较接近，则可一步 步 地 参 照执行。要是预计完成时间落后于要求，或者在力量部署上有忙闲不均，不够紧凑，甚至某些任务超出了实际可能的情况，这就需要在技术和力量的组织方面进行调整。每进行一次调整，就要根据调整后得到的数据进行一次计算，把各种方案加以比较，选取最好的方案，这就是关键路径法的最优化了。这种事先在“图”上进行多次模拟，从而判断哪种做法效益最优的方法可以防患于未然，使计划工作具有一定的科学预见性，大大减少由于临时发现问题而造成人力、财务、物料和时间上不可挽回的损失。

在关键路径法中，上面我们用点代表任务，用有向边代表前后关系，这就是点权网络。另有一种表示法是边权网络，就是用有向边代表任务，而用点来代表任务完成后的结果。当然，对于大多数规划，两种表示法将有不同的网络。有可能由一个给出的网络导出另外一网络。在一般的关键路径法中所画的网络常用边权法，可是前面所讲的点权法在判辨关键路径方面仍是关键路径法。

事实上，点权法有某些优点，它不需要设置虚设任务来保持前后关系，而在边权法中有时是必需的。在点权法中，只有α和ω是虚设任务，而这些有时也不是必要的，如 图 13—20中就没有α和ω。另外，从点权法所画的网络变换到另一种网络是容易得多，而且在点 权 法中，4 个量ES、EF、LS、LF更能够直接得到。

图13—20是图13—14的边权法表示的例子。为了得到一个网络，不再需要虚设任务α 和ω。

源点代表“工程准备开始进行”。因为任务 1 不需要事前准备工作，图中由源点画出一

条有向边，它的终点表示“任务 1 完成，准备进行以后的任务 2，3 和 7 了”。接下去可以看出，任务 9 和11的准备工作只需要任务 2，但是任务 4 的准备工作是任务 2 和 3，所以必须增加二条虚设边β_1和β_2。如果不加虚设边而把任务 2 和 3 画成两条平行有向边，那么就意味着对任务 9 和11也必须先有 2 和 3，这是不符合实际情况的。余类推。

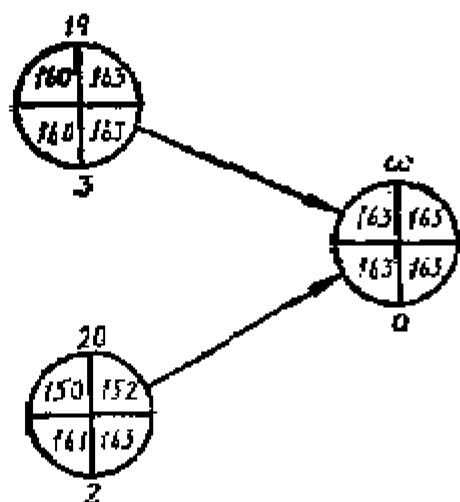


图 13—19

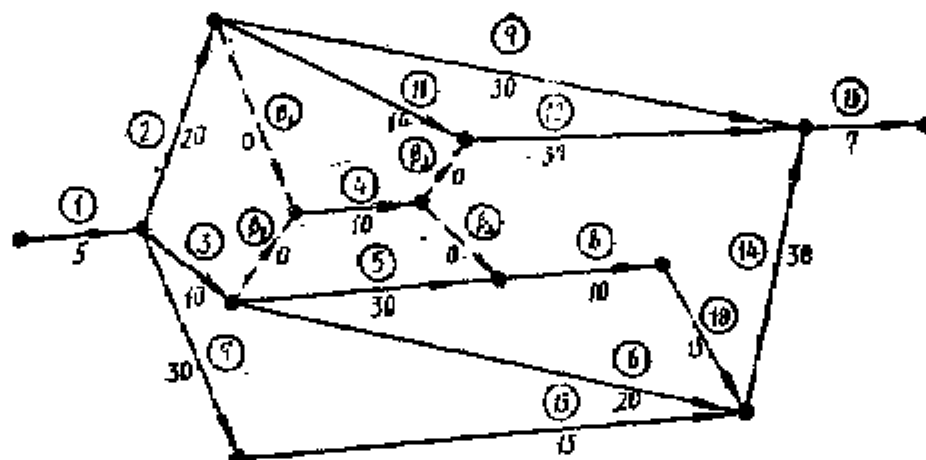


图 13—20

在边权图中，工作时间现在变成有向边的权，所有关键边成为由源点到收点之间的关键路径。关键路径上工作时间的总和是最大的，所以关键路径法在边权法中就成为求最长路径了。

习　　题

13—1　求图13—21中所示的点权交通网络由点 1 到点 4 的极限负载量。

13—2　求图13—22中所示点权交通网络由点 1 到点 5 的极限负载量。

13—3　求图13—23所示的有向点权交通网络由点 1 到点 4 的极限负载量。

13—4　求图13—24中所示的两个有向点权交通网络由点 1 到 5 的极限负载量。

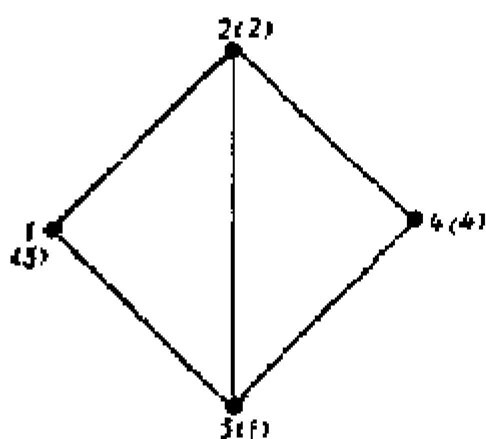


图 13—21

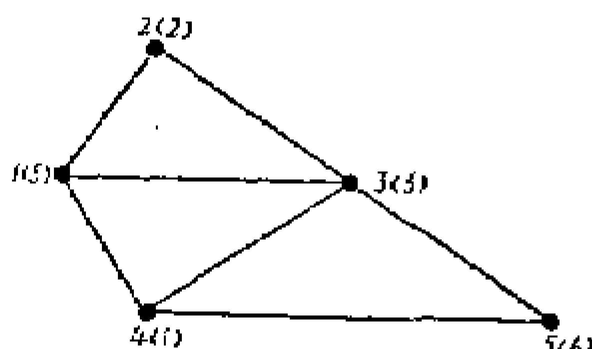


图 13—22

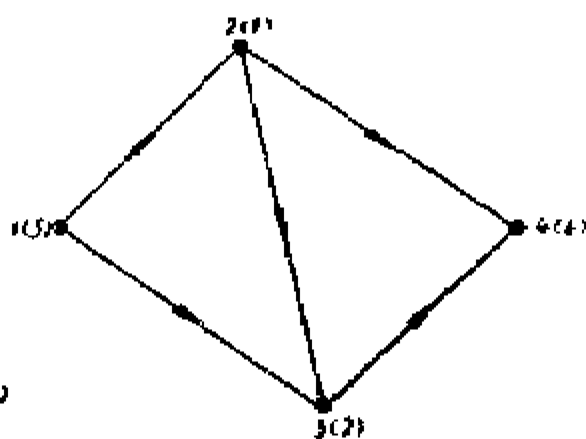


图 13—23

13—5　有一工程划分为下列诸任务：

任务编号	先行任务	工作时间（以天为单位）	任务编号	先行任务	工作时间（以天为单位）
1		6	6	3	3
2		4	7	4，5	10
3	2	5	8	7	12
4	2	6	9	6，8	4
5	2	4			

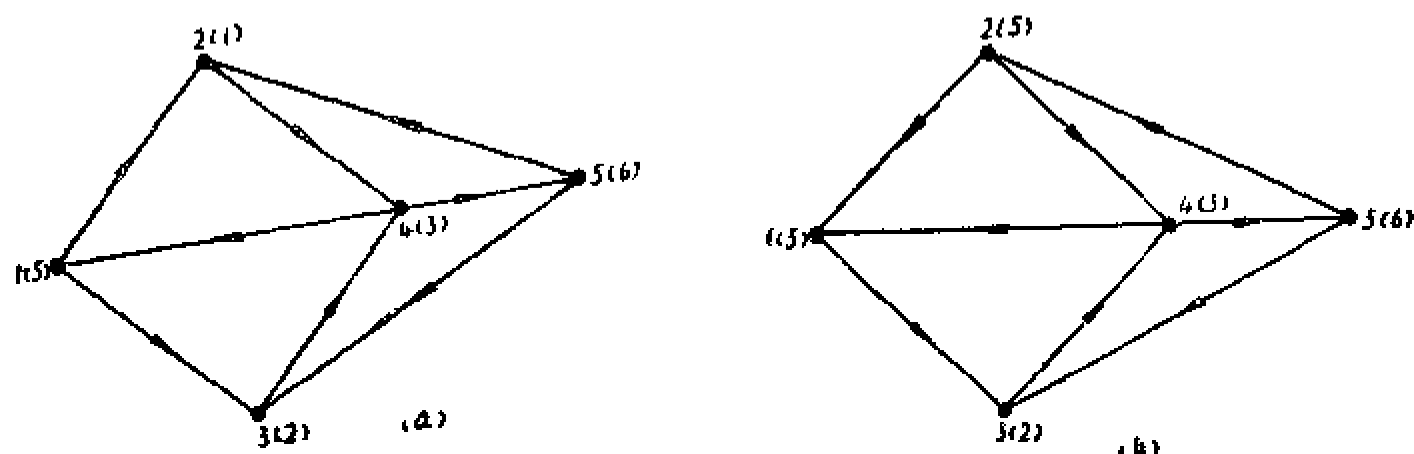


图 13—24

试画出这项工程的任务网络，并计算出每一任务的ES_i，LS_i，EF_i和LF_i。这项工程的关键任务是哪些？这项工程的最早完成时间是多少天？每点的TF_i和FF_i各是多少？

13—6　设习题13—5中任务4的工作时间由6天缩短为4天，则对于上一习题中各计算的时间有何影响？

13—7　试用你所熟悉的程序语言编制出习题13—5的计算程序，验算出关键任务是哪些？各点的ES_i，LS_i，EF_i，LF_i，TF_i和FF_i？

第十四章 故障诊断

第一节 引 言

通常，当一个系统工作时，若系统表现出与设计的功能不一致，导致了错误的结果时，人们便说系统出了故障。

一个系统不能正常工作的原因很多，例如元件、插件、子系统的故障，还有设计上的错误等等。在这一章里，第二节、第三节、第四节考虑系统的诊断的问题，第五节针对接点开关网络中存在的一种设计错误——迂回电路，给出利用图论方法进行诊断的理论。

第二节 分 辨 率

我们这里所考虑的系统是指那些可以用方块和联络线来代表整个系统的工作，其中一个方块完成某一种已知的作用，而联络线传输方块的信息。计算机配置、计算机程序和控制系统等就是一些例子。用点来代表方块，用线来代表联络线，我们可以得到一个对应这些系统的线性图。例如，一部计算机的一个微程序控制部件如图14—1（a）所示，可以用一个有向线性图如图14—1（b）来表示。

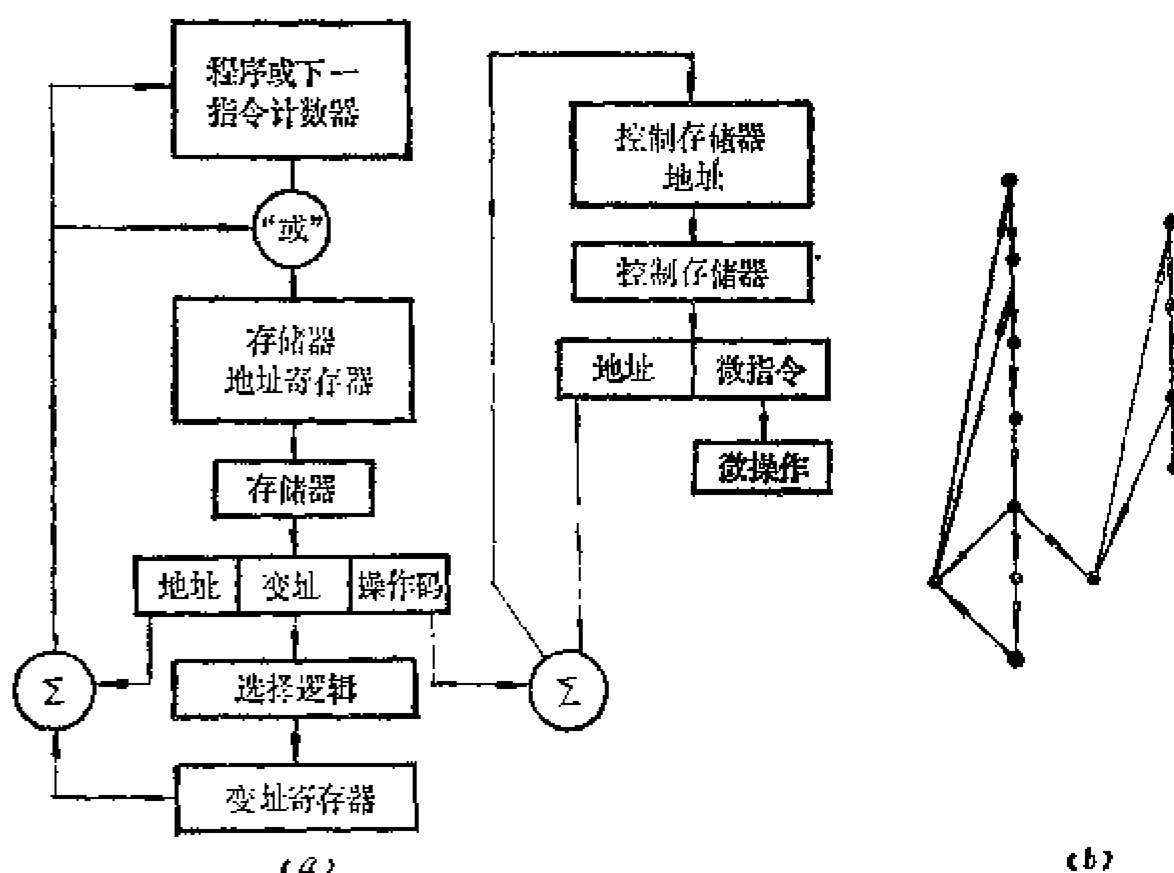


图 14—1

在铁路电气集中联锁系统中办理一条进路时的整个动作层次可由图14—2（a）的方块图来描述。它同样可抽象为一个有向图，如图14—2（b）所示。

信息进入系统的一端叫作输入端，信息从系统中取出的一端叫作输出端。假设每一个输

入（或输出）属于一个方块，我们可以规定一个输入点作为对应方块只有输入端的那个点。同样，一个输出点是对应方块只有输出端的那个点。相应地在一个线性图中，外来信号可以进入的那个点叫作输入点，而信号可以离去的那个点叫作输出点。

当一个系统中的一些方块发生故障，我们作下列假定：

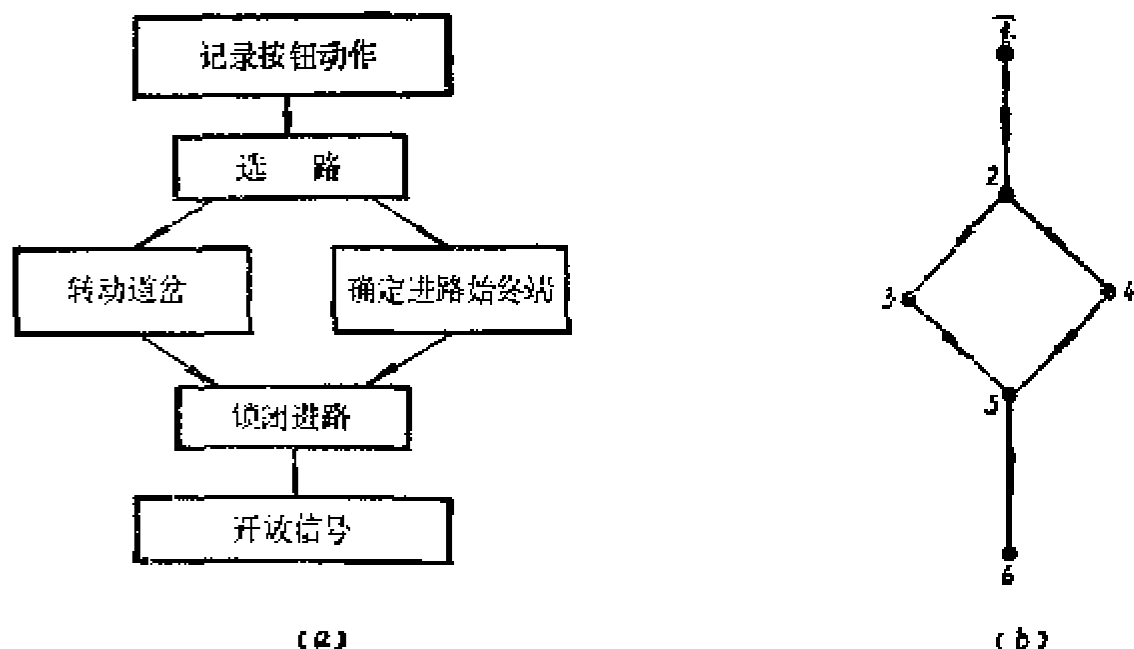


图 14—2

当一个无畸变信号或是一个畸变信号通过一个故障点时，这个信号将是畸变的。另一方面，当一个无畸变（畸变）信号通过一个无故障点时，这个信号将是无畸变（畸变）的。

所谓一个故障点就是对应一个故障方块的点；所谓一个无故障点就是对应一个正常作用方块的点。

有了上述假定，我们知道信号一经畸变，就不能是无畸变的。所以监视一个输出信号，我们可以找出这个信号有否通过一个故障点。实际上，这个假定不适用于所有情况。另一方面，有了这个假定，许多诊断方面的理论结果是容易得到的。而且，这些结果可以修改来包括假定不适用的特殊情况。为了表明哪些点影响一个到达某一特殊端的信号，我们先要引用两个术语。

如果一个有向图满足下列三个条件：

（一）除点 i 和 j 外，对于其它各点，$d^+(v)=d^-(v)$。

（二）$d^+(i)=d^-(i)+1$。

（三）$d^-(j)=d^+(j)+1$。

这个有向图叫作$M(i\times j)$型有向M图。

设 i 和 j 是一个有向图G中的两个点，和符号$V(i\times j)$表示一个点组。当且仅当有$M(i\times j)$型的含有点p的连通有向M图时，点p是在$V(i\times j)$之中；这个点组$V(i\times j)$叫作可测组。

例14—1　在图14—3中有一个有向图，一个可测组$V(1\times 3)$是（1，2，3，4）。这是因为从点1到点3的有向路径$P_1=(a, b)$通过点1，2和3，从1到3的另一条有向路径$P_2=(d, e)$通过点1，3和4。但是没有$M(1\times 3)$型含有点5的连通有向M图。一个可测组$V(1\times 4)$是（1，2，4）。另一个可测组$V(2\times 5)=(2,3,4,5)$。

现在再引用一个符号$\overline{V(i\times j)}$。

设V是一个有向图中所有点的一个点组，那么

$$\overline{V(i\times j)}=V\ominus V(i\times j) \tag{14—1}$$

例如图14—3中有向图中的$\overline{V(1\times 4)}$为（3，5）。

现在来看一看一个可测组$V(i\times j)$不含有一个有向图G中所有点对的情况。假使我们从点i注入一个信号。如果在点j监视这个信号而发现存在有些故障点时，那么这些故障点一定是在$V(i\times j)$之中。另一方面，如果在点j监视到的信号表明没有故障点，而我们知道曾存在过故障点，那么这些故障点很明显是在$\overline{V(i\times j)}$之中。

假使有一个信号注入点i时，在点j监视时表明没有故障点，而在点k监视时却表明有故障点时，那么我们知道故障点存在于组

$$\overline{V(i\times j)}\cap V(i\times k) \qquad (14-2)$$

之中。

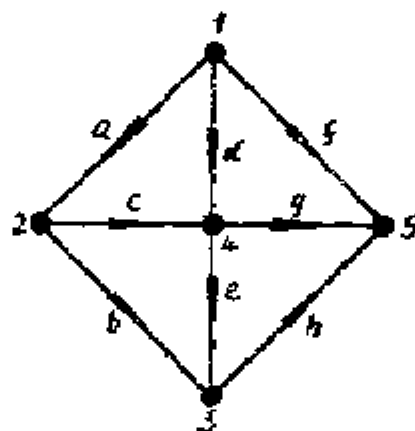


图 14—3

如果信号在点i注入而我们只能在点j和k进行信号监视测试，并且如果有一个故障点的话，那么我们可以在下列诸组中决定故障点在哪一组之中：

$$V(i\times j)\cap V(i\times k)$$
$$V(i\times j)\cap \overline{V(i\times k)}$$
$$\overline{V(i\times j)}\cap V(i\times k)$$
$$\overline{V(i\times j)}\cap \overline{V(i\times k)}$$

这些组形成一个集体叫作可测组$V(i\times j)$和$V(i\times k)$的D-划分，符号为$\{D\}$。D-划分的定义为：

对于给出的k个可测组$V(i_p+j_p)$，$P=1，2，\cdots，k$，下列一些组的集体叫作这些可测组的D-划分$\{D\}$，即

$$\{D\}=\bigcap_{P=1}^{K}\overline{V(i_p\times j_p)} \qquad (14-3)$$

表示$\overline{V(i_p\times j_p)}$的所有可能组合，$p=1，2，\cdots，k$，其中$\overline{V(i_p\times j_p)}$或是$V(i_p\times j_p)$或是$\overline{V(i_p\times j_p)}$。

例14—2　以图14—3中的有向图为例，它的可测组是：

$$V(1\times 3)=(1,2,3,4)$$
$$V(1\times 4)=(1,2,4)$$
$$V(2\times 5)=(2,3,4,5)$$

一个D-划分包括

$$V(1\times 3)\cap V(1\times 4)\cap V(2\times 5)=(2,4)$$
$$V(1\times 3)\cap V(1\times 4)\cap \overline{V(2\times 5)}=(1)$$
$$V(1\times 3)\cap \overline{V(1\times 4)}\cap V(2\times 5)=(3)$$
$$V(1\times 3)\cap \overline{V(1\times 4)}\cap \overline{V(2\times 5)}=\phi$$
$$\overline{V(1\times 3)}\cap V(1\times 4)\cap V(2\times 5)=\phi$$
$$\overline{V(1\times 3)}\cap V(1\times 4)\cap \overline{V(2\times 5)}=\phi$$
$$\overline{V(1\times 3)}\cap \overline{V(1\times 4)}\cap V(2\times 5)=(5)$$
$$\overline{V(1\times 3)}\cap \overline{V(1\times 4)}\cap \overline{V(2\times 5)}=\phi$$

一个D-划分有一个特性如下。

定理14—1：可测组的一个集体的一个D-划分中，任何两个非零组是没有共同点的。

证：我们注意到$V(P\times q)$和$\overline{V(P\times q)}$是没有共同点的。根据D-划分的定义，在一

个D-划分中的任何两组R_1和R_2至少有一个可测组$V(p\times q)$，使$V(p\times q)$是在R_1的表达式中，和$\overline{V(p\times q)}$是在R_2的表达式中，因此R_1和R_2是没有共同点的。

一个可测组$V(p\times q)$可以看作是相当于在点p注入一个信号，而在点q监视信号那样进行一次测试。所以如果在一个D-划分$\{D\}$中的一个组包括一个点v，又如果只有点v是有故障的，那么我们根据从所给出的可测组集体所得到$\{D\}$而进行测试时，可以确定点v是故障点。可是另一方面，如果两个点v_1和v_2都在可测组的集体M下的一个D-划分中的同一个组内，又如果这两个点之一是有故障的，那么只靠对应M中可测组的测试来确定究竟哪一个点是有故障是不可能的。

在一个有向图的可测组的一个集体M中，如果（1）在M的一个D-划分中有一组R，它含有k个点，和（2）M的D-划分中没有一个组含有k个以上的点，那么这个有向图叫作有k-分辨率的，或叫作是k-可分辨的。

例如，在图14—3的有向图中，在可测组$V(1\times 3)$，$V(1\times 4)$和$V(2\times 5)$的集体M中，因为在M的D-划分$\{D\}$中有一个组有两个点，即（2，4），而没有其它组多于两个点的，所以这个有向图在可测组$V(1\times 3)$，$V(1\times 4)$和$V(2\times 5)$的集体M中有2-分辨率。

注意在可测组集体M中，一个有向图G是k-可分辨的意思是说有k个点的一个组，如果其中有一个点是有故障的，对于这些可测组的测试不能确定k个点中哪一个点是有故障的，但是这些测试却将表明k个点中含有一个故障点。

第三节　测　试　点

我们在前一节中已经对一个系统的输入端和输出端下过定义。一个系统中任何其它端统称为内部端。就是说，一个内部端是方块的一个部分。有一根联络线，但是它既不是输入端，也不是输出端。所以对应一个系统的线性图中的每一个点表示出它是一个输入端，或是一个输出端，或者一个内部端。

对于测试而言，如果一个系统中的任何一个内部端可以是一个输入端或是一个输出端，那么在任何一个可测组$V(p\times q)$中的一些点p和q可以用作诊断用的可测组集体的一个组成部分。即使所有点可以是输入点和输出点，可是1-分辨率还是不可能的。例如图14—4中的有向图G，在G中所有点p和q的可测组$V(p\times q)$的一个集体的情况下，不是1-可分辨的。为了排除这些困难，我们用测试门的办法在需要时删除一些边。关于测试门的特性以及如何用测试门来断诊一个系统的方法将在下一节中再研究。

另外一种关于给出的k的k-分辨率变成不可能的明显情况是当输入点和输出点不够的时候。在这种情况时，我们用所谓测试点来得到另外可测组，这样使得k-分辨率成为可能。换句话说，我们想把测试点看作是在一条边上的附加的点，为了诊断，我们可以注入也可以监视一个信号。可是当我们考虑G中一些点的分辨率时，我们不想把一个插入点看作是一个给出图G中点组的一个元素。所以一个测试点的定义为：

指定给一条边的一个测试点是打算把一个信号注入这条边，并且对通过这条边的信号加以监视。

的。例如，在图14—5中，$v(a^+)=1$，$v(a^-)=2$。同样，$v(b^+)=2$，$v(b^-)=3$。

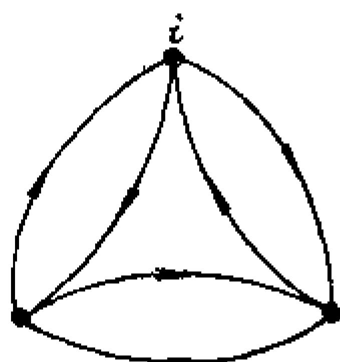


图 14—4

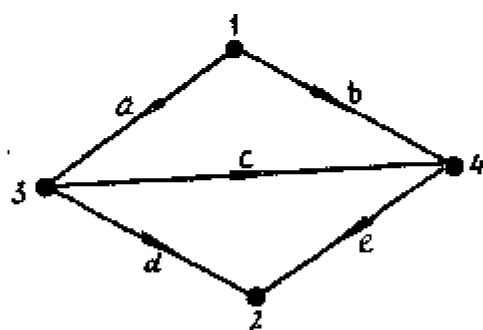


图 14—5

现在再介绍几个符号：

符号$V(a^{\pm}\times j)$是$V[v(a^{\pm}\times j)]$的缩写记号。同样，$V(i\times a^{\pm})$是$V[i\times v(a^{\pm})]$的缩写记号，$V(a^{\pm}\times b^{\pm})$是$V[v(a^{\pm})\times v(b^{\pm})]$的缩写记号。这种缩写记号可以用于有向$M$图，如$M(a^{\pm}\times b^{\pm})$表示$M[v(a^{\pm})\times v(b^{\pm})]$。

例14—3 图14—5的有向图G只有一个输入点1和只有一个输出点2。所以我们只有一个可测组$V(1\times 2)=(1,\ 2,\ 3,\ 4)$。如果在边$e$上指定一个测试点，那么我们就有组$V(1\times e^+)$和$V(e^-\times 2)$作为外加可测组。

在研究测试点的特性时，我们把只有一个输入点和一个输出点的连通有向图叫作单输入单输出连通图。

设i和j分别为一个单输入单输出连通图的输入点和输出点。因为从点i注入的信号只能在点j得到，所以任何不在可测组$V(i\times j)$中的一个点所代表的方块，对于单输入单输出连通图所表示的系统的性能是无所供献的。所以我们只考虑一个单输入单输出连通图的可测组包括这个单输入单输出连通图中所有的点。

设一个从i到j的单输入单输出连通图，它的输入点为i，它的输出点为j。

下面我们首先研究测试点与有向截割组的关系。

定理14—2：设G是从i到j的一个单输入单输出连通图，假定$S(e)$是一个分开i和j并含有边e的有向截割组。设g_1和g_2是从G中删除$S(e)$中所有边后所得到的两个部分。那么$V(i\times e^+)=V(g_1)$和$V(e^-\times j)=V(g_2)$。

证：设p是g_1的一个点。假设有一个$M(e^-\times j)$型的连通有向M图，它含有点p。那么因$S(e)$中至少有一条边的定向是从$V(g_2)$中的一个点到$V(g_1)$中的一个点的，$S(e)$不是一个有向截割组。所以没有一个$M(e^-\times j)$型的有向M图含有$V(g_1)$中的一个点。

假使p不在任何$M(i\times e^+)$型的连通有向M图中。设V_p是所有点的一个点组，这样对于在V_p中的任何点v'，在g_1中就有一条从p到v'的有向路径。因为$V(i\times j)$含有点p，就没有一个$M(i\times e^+)$型的有向M图通过V_p中的点v'。注意点$v(e^+)$不在V_p之中。我们也可以知道点i不在V_p之中。设想有一个截割组$S'(e)$

$$S'(e)=E\{[V(g_1)\ominus V_p]\times[\overline{V(g_1)\ominus V_p}]\}\,U\,E\{[\overline{V(g_1)\ominus V_p}]\times[V(g_1)\ominus V_p]\} \quad (14—4)$$

因为$S(e)$是一个有向截割组，$E[\overline{V(g_1)}\times V(g_1)]$是空的，所以$E\{V(g_1)\times[V(g_1)\ominus V_p]\}$是空的。在$V_p$中的一个点$v'$与$[V(g_1)\ominus V_p]$中的一个点$v''$之间所连接的一条边的定向不会是从$v'$到$v''$的。这是因为如果是这样的话，$v''$就一定在$V_p$之中。所以$E\{V_p\times[V(g_1)\ominus V_p]\}$是空的。同样$E\{[V(g_1)\ominus V_p]\times[V(g_1)\ominus V_p]\}$也是空的。这就是说$S'(e)$是分开$i$和$j$的

一个有向截割组，这是一个矛盾。因为根据假定只有$S(e)$是含有边e的有向截割组。因为点p一定在至少一个$M(i\times e^{+})$型连通有向M图中。同样，我们可以取G_2中的一个点证明这个定理是正确的。▮

当有一个以上的含有边e并分开i和j的有向截割组时，下面定理是适用的。

定理14—3：设G是从i到j的一个单输入单输出连通图。假使在所有分开i和j的有向截割组$S_1(e)$，$S_2(e)$，…，$S_K(e)$中是含有边e的，那么$V(i\times e^{+})=V(g_1)$和$V(e^{-}\times j)=V(g_2)$，其中g_1和g_2是从G中删除$S_1(e)$，$S_2(e)$，…，$S_K(e)$中所有边后所得到的两个部分。

证：对于g_1和g_2中任何点的证明正像定理14—2的证明一样，所以我们只需要证明既不在g_1中又不在g_2中的任何点不会在$V(i\times e^{+})$中或$V(e^{-}\times j)$中。设想一个子图g_3是在删除g_1和g_2中所有点以及连接到这些点的所有边得到的。

我们可以容易地知道，在$S_1(e)$，$S_2(e)$，…，$S_K(e)$中只有两个截割组，在删除了这两个截割组的所有边后会得到3个子图g_1，g_2和g_3。设这两个截割组是$S_1(e)$和$S_k(e)$。再设v'是g_3中的一个点。如果v'是在$M(i\times e^{+})$型的一个连接有向M图中，那么$S_1(e)$和$S_k(e)$两者之中就有一个不是有向截割组，因此v'不能在$M(i\times e^{+})$型的任何一个连通有向M图中。同样，我们可以证明v'不能在任何一个$M(e^{-}\times j)$型的连通有向M图中。因此这个定理是正确的。▮

在定理14—3的证明中，我们用两个有向截割组$S_1(e)$和$S_k(e)$，使在删除这两个截割组中所有边后得到g_1、g_2和g_3。我们说这两个有向截割组$S_1(e)$和$S_k(e)$由边e覆盖，或者说边e覆盖了$S_1(e)$和$S_k(e)$。当像定理14—1中只有一个截割组$S(e)$时，我们就说边e覆盖有向截割组$S(e)$。覆盖的正式定义为：

如果一个有向截割组S是分开i和j且含有边e的准一截割组，那么边e叫作覆盖着S。当且仅当（1）有向截割组$S_1=E(V_1\times\overline{V_1})$和$S_2=E(V_2\times\overline{V_2})$分开$i$和$j$，（2）$S_1$和$S_2$含有边$e$，和（3）其它任何一个分开$i$和$j$且含有边$e$的截割组$S'=E(V_3\times\overline{V_3})$满足下列条件：

$$(V_1\cap V_2)\supset V_3\supset(V_1\cap V_2)\qquad(14—5)$$

时，边e覆盖着S_1和S_2。

例14—4　图14—6（a）中有一个从1到6的单输入单输出连通图。$S(b)$是分开1和6且含有边b的唯一有向截割组。所以边b覆盖有向截割组$S(b)$。删除$S(b)$中所有边后，我们得到g_1和g_2如图（b）所示。因此$V(1\times b^{+})=V(g_1)$和$V(b^{-}\times 6)=V(g_2)$。$S_1(h)$，$S_2(h)$和$S_3(h)$是分开点1和6且含有边h的所有有向截割组。由图（a）可见

$$S_1(h)=E(V_1\times\overline{V_1})=E\{(1,2,4)\times\overline{(1,2,4)}\}$$

$$S_2(h)=E(V_2\times\overline{V_2})=E\{(1,2,3,4,5)\times\overline{(1,2,3,4,5)}\}$$

$$S_3(h)=E(V_3\times\overline{V_3})=E\{(1,2,3,4)\times\overline{(1,2,3,4)}\}$$

所以

$$\{V_1\cup V_2=(1,2,3,4,5)\}\supset\{V_3=(1,2,3,4)\}\supset\{V_1\cap V_2=(1,2,4)\}$$

因此边h覆盖着$S_1(h)$和$S_2(h)$。子图g_1，g_2和g_3如图（c）所示。

注意：　$V(1\times h^{+})=V(g_1)$和$V(h^{-}\times 6)=V(g_2)$

定理14—2和14—3的逆定理也是正确的。

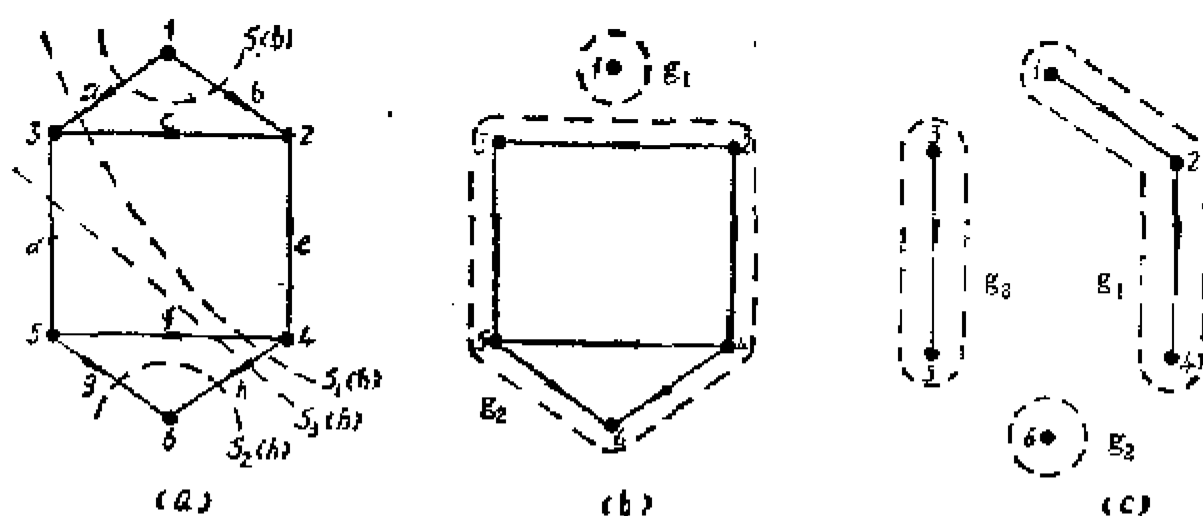


图 14—6

定理14—4：设G是从i到j的一个单输入单输出连通图。设$V(i\times e^+)$和$V(e^-\times j)$为可测组。如果$V(i\times e^+)\cap V(e^-\times j)=\phi$，那么就会有一个含有边$e$的有向截割组$S(e)$和$S'(e)$，使$V(g_e)=V(i\times e^+)$和$\overline{V(g_e)}=V(e^-\times j)$，其中

$$S(e)=E\{V(g_e)\times\overline{V(g_e)}\}$$

$$S'(e)=E\{V(g'_e)\times\overline{V(g'e)}\}$$

证：有一个边组$S=E\{V(i\times e^+)\times\overline{V(i\times e^+)}\}$如图14—7所示。如果边$e$不在$S$中，那么$V(i\times e^+)$含有点$v(e^-)$。既然根据假定，$V(e^-\times j)$含有点$v(e^-)$，我们有$V(i\times e^+)\cap(e^-\times j)=\phi$，所以边$e$一定是在$S$之中。设$S=(b_1, b_2, \cdots, b_m, e)$。假定$S$不是一个有向截割组，那么至少有一条边，比如说边$b_1$的定向是从$V(e^-\times j)$中的一个点$V(b^+)$到$V(i\times e^+)$中的一个点$v(b_1^-)$的。由于下列诸理由这是不可能的。

（一）因为$v(b_1^-)$是在$V(i\times e^+)$之中，所以至少有一个含有点$v(b^-)$的$M(i\times e^+)$型的连通有向M图。

（二）因为G是一个从i到j的单输入单输出连通图，所以至少有一条从i到$v(b_1^+)$的有向路径。

（三）因此有一条从i到$v(e^+)$且含有边b_1的有向路径。这就是说$v(b^+)$一定是在$V(i\times e^+)$之中。这是矛盾的。换句话说，S是一个含有边e的有向截割组。而且，删除S中所有边后将会给出一个部分g_e，它包含$E[V(i\times e^+)\times V(e^-\times j)]$中的那些边。同样，我们可以用$V(e^-\times j)$证明有一个有向截割组$S'(e)$。

如果在定理14—4中$V(i\times e^+)\cap V(e^-\times j)=\phi$的条件不能满足，那么下一定理示出不会有一个含有边$e$的分开$i$和$j$的有向截割组。

定理14—5：设G是一个从i到j的单输入单输出连通图。假使可测组$V(i\times e^+)$和$V(e^-\times j)$有下列特性：

$$V(i\times e^+)\cap V(e^-\times j)=\phi$$

那么就不存在含有边e的分开i和j的有向截割组。

证：设v是在$V(i\times e^+)$和$V(e^-\times j)$中的一个公共点，那么就存在着一个含有点v的$M(i\times e^+)$型的连通有向M图，也存在着一个含有点v的$M(e^-\times j)$型的连通有向M图，如图14—8所示。所以存在着一条含有边e的有向路径。因此就没有含有边e的有向截割组。▮

一个重要问题是如何选择些边使把测试点指定给这些边时，可以达到1-分辨率。为了了解这个问题，先对几个术语下定义。

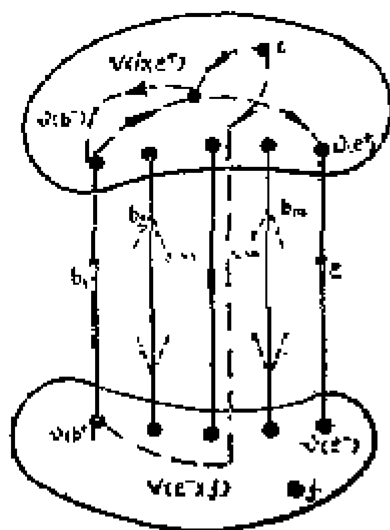


图 14—7

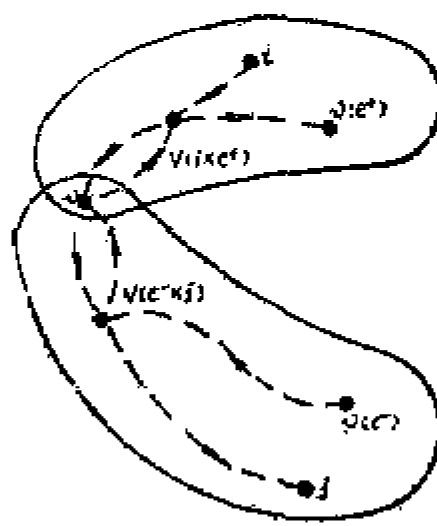


图 14—8

对于一个边组E，所谓在E下的可测组是指当我们把测试点指定给E中所有边时的所有可能的可测组。

所谓一个从i到j的单输入单输出连通图在边组E下是1-可辨的是指当测试点指定给E中每一条边时，在E下的那些可测组将是1-可分辨的。

例如图14—6中的一个从点1到6的单输入单输出连通图G中，假使边组E包括边a和h，那么在E下的可测组为

$$V(1\times a^-)=(1)$$
$$V(a^-\times 6)=(3,5,6)$$
$$V(1\times h^-)=(1,2,4)$$
$$V(h^-\times 6)=(6)$$
$$V(a^-\times h^+)=\phi$$
$$V(h^-\times a^+)=\phi$$

因为点3和5对这些可测组来讲，都在D-划分的一个组中，所以G不是1-可辨的。

下面一个定理给出不能达到1-分辨率的边组E的一个条件。

定理14—6：设G是从i到j的一个单输入单输出连通图，又设E是一个边组，和$\{S\}$是E的那些边覆盖下的所有有向截割组的一个集体。那么如果在G中存在着一条边不在$\{S\}$中任何一个有向截割组的话，在E下G不是1-可分辨的。

证：情况一，假使有一对可测组，而$V(i\times b^+)\cap V(b^-\times j)=\phi$，那么根据定理14—5就有一个含有边$b$的有向面路$C$。我们可以知道如果$C$中任何一条边是在$M(i\times e^+)$或$M(e^-\times j)$型的一个连通有向$M$图中的话，那么就有另一个同型的连通有向$M$图含有$C$中所有边，其中$e$是$E$中的一条边。所以或者是属于$C$的所有点都在一个可测组中，或者属于$C$的那些点没有一点是在一个可测组中。因此在$E$下对于一些可测组$D$-划分中的一个组含有属于$C$的所有点，并且$G$在$E$下不是1-可分辨的。因为$C$中的那些边不能在任何一个有向截割组中，所以这个定理在情况一是正确的。

情况二，假使对于E中每条边e，$V(i\times e^+)\cap V(e^-\times j)=\phi$。设$\{S\}$是有向截割组$S_1$，$S_2\cdots$，$S_n$的一个集体。又设$S_p=E(V_p\times\overline{V}_p)$，$p=1$，$2$，$\cdots$，$m$。根据定理14—4，对于任何一个可测组$V(i\times e^+)$和$V(e^-\times j)$，在$\{S\}$中有一个有向截割组$S_k$，而$V(e^-\times j)=\overline{V}_k$。因为至少一条边，譬如说边$b$不在$\{S\}$中的任何一个有向截割组中，点$v(b^+)$和$v(b^-)$两者一定或者都在$V_p$之中，或者都在$\overline{V}_p$之中，$p=1$，$2$，$\cdots$，$m$。所以没有一个可测组含有两者之一，即$v(b^+)$或$v(b^-)$。因此在有$E$的可测组的集体的情况下，在$D$-划分

的一个组中如果含有$v(b^-)$，也将含有$v(b^+)$。所以根据k-分辨率的定义，G在E下不是1-可分辨的。▮

例14—5　在图14—9中有一个从点1到5的单输入单输出连通图。假使$E=(a,b)$，那么S包括：

$$S_1=E\{(1)\times(\overline{1})\}$$
$$S_2=E\{(1,2)\times(\overline{1,2})\}$$
$$S_3=E\{(1,2,3,4)\times(\overline{1,2,3,4})\}$$

注意截割组S_1由边a覆盖，截割组S_2和S_3由边b所覆盖。因为边e不在$\{S\}$中任何一个有向截割组之中，G在E下不是1-可分辨的。为了检查这个结果，我们首先要得到所有可能的非零可测组：

$$V(1\times5)=(1,2,3,4,5)$$
$$V(1\times a^+)=(1)$$
$$V(a^-\times5)=(2,3,4,5)$$
$$V(1\times b^+)=(1,2)$$
$$V(b^-\times5)=(5)$$
$$V(a^-\times b^+)=(2)$$

从这些可测组中，我们可以得到D-划分如下：

$$V(1\times5)\cap\overline{V(1\times a^+)}\cap V(a^-\times5)\cap\overline{V(1\times b^+)}\cap\overline{V(b^-\times5)}\cap$$
$$\overline{V(a^-\times b^+)}=(1,2,3,4,5)\cap(2,3,4,5)\cap(2,3,4,5)\cap(3,4,5)$$
$$\cap(1,2,3,4)\cap(1,3,4,5)=(3,4)$$

因此在E下，G不是1-可分辨的。

定理14—6表明如果有一条边不在$\{S\}$中的一个截割组中的话，一个单输入单输出连通图G不是1-可分辨的。如果每条边是在$\{S\}$中的一个截割组中，那么G是否是1-可分辨的呢？答案一般是否定的。一个例子是图14—10中的单输入单输出连通图G，其中$E=(a)$，就是说，$\{S\}$包括S_1和S_2含有G中的所有边。

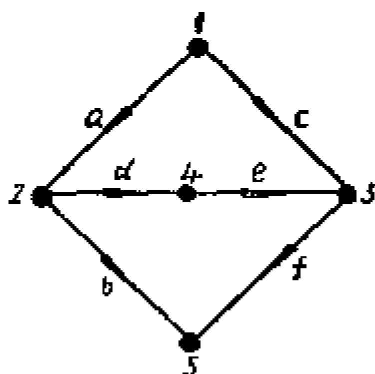


图　14—9

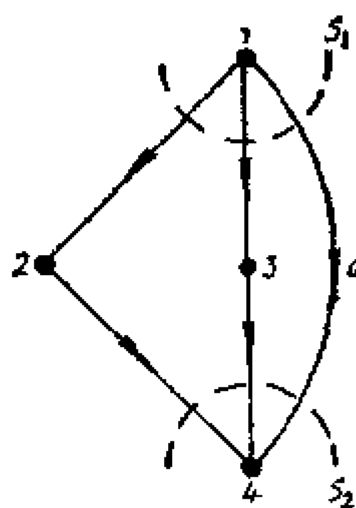


图　14—10

可是，可测组为：

$$V(1\times4)=(1,2,3,4)$$
$$V(1\times a^+)=(1)$$
$$V(a^-\times4)=(4)$$

因此在这些可测组中，点 2 和 3 都在一个D-划分中的一个组中，就是说在E下G不是1-可分辨的。

进一步研究可测组可能启发我们关于1-分辨率的问题。

定理14—8：设G是一个从 i 到 j 的单输入单输出连通图，E是一个边组，设$\{V\}$是由于对E中所有边指定了测试点后所产生的所有可测组的一个集体。那么当且仅当对于G中每一对点v_r和v_s有一个可测组$V\in\{V\}$，和V只含有v_r或v_s但不是两者时，G在E下是1-可分辨的。

证：设v_0，v_1，…，v_n是G中的所有点，再设想一个点v_0，那么对于G中另一个点v_p有一个可测组$V_p=\in\{V\}$，且v_p只含有v_0或v_1，但不是两者。因为这对于G中所有点是正确的，在E下就存在着D-划分中的一个组D有下列关系：

$$D=\widetilde{V}_1\cap\widetilde{V}_2\cap\cdots,\ \cap\widetilde{V}_m=\{v_0\}$$

而且，这对于G中每个点都是正确的。因此一个D-划分中每一个非零组包括一个点，根据定义，它是1-可分辨的。

相反，如果G在E下是1-可分辨的，那么一个D-划分中每一个非零组包括一个点，所以对于一对点v_r和v_s，在E下在一个D-划分中有一个组D，它包括v_r，其中

$$D=\widetilde{V}_1\cap\widetilde{V}_2\cap\cdots\cap\widetilde{V}_m \qquad (14-6)$$

其中一定有V_r（$1\leqslant r\leqslant m$）只含有v_r或v_s，但不是两者。因此在$\{V\}$中有一个可测组V_r，它只含有v_r或v_s但不是两者。▌

因为得到一个分开i和j并且由E中的那些边所覆盖的所有可能的有向截割组的一个集体$\{S\}$比得到所有可能的可测组的一个集体$\{V\}$为较容易，所以我们把定理14—7另行陈述如下：

定理14—8：设G是从 i 到 j 的一个单输入单输出连通图，和E是一个边组。设$\{S\}$是分开 i 和 j 并且由E中那些边所覆盖的所有可能的有向截割组的一个集体。当且仅当对于任何一对点v_r和v_s存在着$\{S\}$中的一个有向截割组分开v_r和v_s时，G在E下是1-可分辨的。

证：从定理14—7我们得到，当且仅当对于任何一对点v_r和v_s存在着一个可测组V，它只含有v_r或v_s但不是两者时，G是1-可分辨的。从定理14—2和14—3可知，一个可测组V可以表达为$V(g')$。如果$V(i\times e^+)\cap V(e^-\times j)=\phi$时，$g'$是在删除在$\{S\}$中的一个有向截割组中所有边后所得到的两部分之一。并且明显地，v_r和v_s中之一是在$V(g')$之中，而另一个是在$\overline{V(g')}$之中。所以我们只要证明$V(i\times e^+)\cap V(e^-\times j)=\phi$对$E$中所有边都能满足。

根据定理14—5，如果$V(i\times e^+)\cap V(e^-\times j)=\phi$，那么在$\{S\}$中没有一个有向截割组含有边 e。另一方面，根据在这个定理中所做的假定，在$\{S\}$中有一个组S分开边 e 的端点$v(e^+)$和$v(e^-)$。所以S是一个含有边 e 的有向截割组。因此$V(i\times e^-)\cap V(e^-\times j)=\phi$这一条件一定满足$E$中每一条边。这就证明了这个定理的必要部分。很明显，这个定理的充分部分是从定理14—7而来的。▌

要找出每一对点是否能由$\{S\}$中的一个有向截割组所分开不是一件容易的工作。因此我们先找寻一个单输入单输出连通图在E下是1-可分辨的一个比较简单些的必要和充分条件。我们首先研究图中各点的次序关系。

如果从v_r到v_s没有有向路径的话，v_r和v_s这两个点的次序排成$v_r>v_s$。例如在图14—9的单输入单输出连通图中，各点的次序为：

$$1>2>4>3>5$$

有了这个各点次序关系的定义，我们可以阐述下面一些定理。

定理14—9：设G是一个从v_1到v_n的单输入单输出连通图，它包括点v_1，v_2，…，v_n。如果G中没有有向回路，那么所有连接到v_1的那些边的定向是出自v_1的，而所有连接到v_n的那些边的定向是朝向v_n的。

证：因为我们只考虑从v_1到v_n的一个单输入单输出连通图，它有一个特性，就是可测组$V(v_1\times v_n)$含有这个图中所有的点，所以就有一条从v_1到G中任何点的有向路径。因此所有连接到v_1的那些边的定向一定是出自v_1的。这样在G中才不致有有向回路。同样，所有连接到v_n的那些边的定向一定是朝向v_n的，这样G就没有有向回路。▌

定理14—10：设G是一个从v_1到v_n的单输入单输出连通图，它包括点v_1，v_2…，v_n。假使G中没有有向回路，那么G中各点可以按次序排列的。

证：对于$n=2$，这个定理很清楚地是正确的。假使这个定理对于$n=k$也是正确的，那么对于$n=k+1$，我们可以从G中删除点v_{k+1}，以及删除连接到v_{k+1}的所有边而组成一个有向图g。设v_k，v_{k-1}，…，v_{k-p}是g中的一些点，所有连接到这些点的定向是朝向这些点的。对于$p=0$，g是一个从v_1到v_k的单输入单输出连通图。因此根据假定，所有在g中的那些点是可以排列次序如$v_1>v_2>\cdots>v_k$的。所以$v_1>v_2>\cdots v_k>v_{k+1}$是所需要的一个次序。

对于$p\geqslant1$，我们在g中插入一些边$y_1,y_2,\cdots,y_p$，其中y_i是从v_{k-i}连接到v_k的一条边，如图14—11所示。这个修改过的有向图是一个从v_1到v_k的单输入单输出连通图。因此它与前图完全相同，所以在G中连同$y_1,y_2,\cdots,y_p$各点是可以依次序排列的。当边$y_1,y_2,\cdots,y_p$删除后，各点的次序关系依然能够满足。因此这个定理是正确的。▌

有了各点的次序关系，我们可以得到一个没有有向回路的单输入单输出连通图的下列一个特性。

定理14—11：设G是一个从v_1到v_n的单输入单输出连通图，它包括n个点而不含有有向回路。那么它就有$n-1$个线性无关的有向截割组，它们分开v_1和v_n。

证：设$v_1>v_2>\cdots>v_n$是根据定理14—10的各点次序，那么截割组S_r

$$S_r=E[(v_1v_2\cdots v_r)\times(v_{r+1}\cdots v_n)]$$

是一个分开v_1和v_n的有向截割组，$r=1，2，\cdots，n-1$。这些截割组明显地是线性无关的。▌

例14—6　图14—12中有一个从点1到6的单输入单输出连通图，各点次序为$1>2>3>4>5>6$。5个线性无关有向截割组为：

$$S_1=E\{(1)\times(2,3,4,5,6)\}=(a,\ b)$$
$$S_2=E\{(1,2)\times(3,4,5,6)\}=(a,c,e)$$
$$S_3=E\{(1,2,3)\times(4,5,6)\}=(d,e)$$
$$S_4=E\{(1,2,3,4)\times(5,6)\}=(d,f,h)$$
$$S_5=E\{(1,2,3,4,5)\times(6)\}=(g,h)$$

注意当有有向回路时，所有各点不能依次序排列。如果从v_1到v_i和从v_i到v_j都没有有向回路时，各点的排列次序就不是唯一的。例如，图14—12中的单输入单输出连通图的另一种各点次序为$1>2>4>3>5>6$。

应用下一定理可能得到各点的排列次序与1-分辨率的关系。

定理14—12：设G是一个从i到j的单输入单输出连通图，它有n个点。设$S_p=E(V_p\times\overline{V}_p)$

是一个分开 i 和 j 的有向截割组，$p=1,2,\cdots,\ n-1$。假使$S_1,\ S_2,\cdots,\ S_{n-1}$是线性无关的。那么对于$G$中任何一对点$v_r$和$v_s$就有$S_r$（$1\leqslant r\leqslant n-1$）把它们分开。

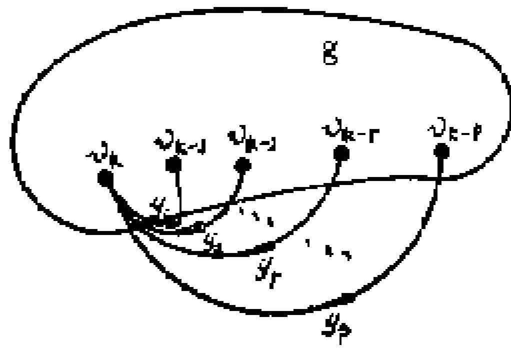


图 14—11

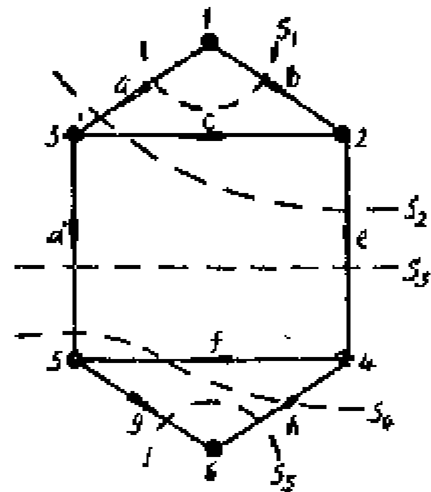


图 14—12

证：假使有一对点v_r和v_s，它们不能由$S_1,S_2,\cdots,S_{n-1}$中任何一组所分开。那么v_r和v_s两者或者都在V_p中，或者都在$\overline{V}_p$中，$p=1,\ 2,\ \cdots,\ n-1$。把v_r和v_s合并，我们得到一个新的有向图G^1。G^1的$S_1,\ S_2,\cdots,S_{n-1}$很清楚是与G的那些截割组相同的。因此$S_1,\ S_2,\ \cdots,\ S_{n-1}$是线性无关的。可是$G^1$只有 $n-1$ 个点。所以不会有$n-2$ 个以上的线性无关截割组的。因此在G中是没有这样一对点的。▌

有了这些结果，我们可以回到1-分辨率这个问题上来了。

定理14—13：设G是一个从 i 到 j 的单输入单输出连通图，其中没有有向回路。假使各点的排列次序是唯一的，那么当且仅当边组E中那些组覆盖着分开 i 和 j 的所有线性无关有向截割组时，G在E下是1-可分辨的。

证：因为各点的排序次序是唯一的，确定着分开 i 和 j 的那些线性无关截割组也是唯一的。如果这些有向截割组中任何一组不能由E中那些边所覆盖，那么图中就有些点不能由E的那些边所覆盖的那些截割组中任何一组所分开。因此根据定理14—8，在E下G不是1-可分辨的。这是与假定相矛盾的。这个定理的逆也可以用与假定相矛盾的道理来证明的。▌

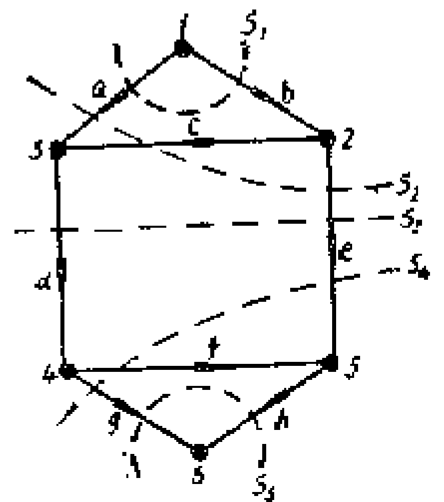


图 14—13

例14—7　图14—13中有一个从点1到6的单输入单输出连通图，因为这个图中各点的排列次序是唯一的，分开1和6的那些线性无关有向截割组是$S_1,\ S_2,\ S_3,\ S_4$和S_5，如图14—13所示。根据定理14—13，为了使这个单输入单输出连通图在E下是1-可分辨的，我们一定在组E中选择些边使这些边覆盖所有这些有向截割组。例如，组E可以是（a，d，g）。

现在我们找出来了分开 i 和 j 的 $n-1$ 个线性无关有向截割组对于1-分辨率的重要性了。如果G在某个边组E下是1-可分辨的，是否在一个单输入单输出连通图G中从 i 到 j 有 $n-1$ 个分开 i 和 j 的线性无关有向截割组呢？答案是"是"的。下一定理给出一个理由。注意由定理14—10知道，如果G没有有向回路，分开 i 和 j 的有 $n-1$ 个线性无关有向截割组。

定理14—14：在一个从 i 到 j 的单输入单输出连通图中，一个有向回路中的顶点是不可分辨的。

证。因为一个有向回路中的那些边不可能包括在一个有向截割组中，所以在一个有向回路中的那些点不能用一个有向截割组来分开的。因此根据定理14—8，这个定理是正确的。▌

根据这个定理，如果一个单输入单输出连通图中有一个有向回路，这个图在一个边组E下不可能是1-可分辨的。因此我们可以说一个从i到j的1-可分辨的单输入单输出连通图会有$n-1$个分开i和j的线性无向有向截割组。

当一个单输入单输出连通图中的点的排列次序是唯一的，定理14—13就给出了一个1-分辨率的简单的必要和充分条件。假使一个单输入单输出连通图中的点的排列次序不是唯一的，那么由组E中的那些边所覆盖的$n-1$个线性无关有向截割组成为一个充分的但不是必要的条件了。

定理14—15：设G是一个从i到j的有n个点的单输入单输出连通图。如果分开i和j的$n-1$个线性无关有向截割组由组E的那些边所覆盖的话，那么G在组E下是1-可分辨的。

这个定理的证明是直接从定理14—12而来的。为了显示上述条件不是必要的，设想有一个从i到j的单输入单输出连通图如图14—14所示。一个边组E选为$E=(a,\ b,\ c)$。一个分开i和j的由E的各边所覆盖的有向截割组集体$\{S\}$，包括S_1，S_2，S_3和S_4：

$$S_1=(d,e,f,a)=E\{(i,v_1,v_2,v_3)\times(v_4,v_5,v_6,j)\}$$

$$S_2=(g,h,k,b)=E\{(i,v_1,v_4,v_5)\times(v_2,v_3,v_6,j)\}$$

$$S_3=(l,m,n,c)=E\{(i,v_3,v_5,v_6)\times(v_1,v_2,v_4,j)\}$$

$$S_4=(a,b,c)=E\{(i,v_1,v_2,v_3,v_4,v_5,v_6)\times(j)\}$$

注意S_1和S_4由边a所覆盖，S_2和S_4由边b所覆盖，S_3和S_4由边c所覆盖。因为任何两个点由这4个有向截割组中的一个组所分开，根据定理14—7这个单输入单输出连通图在E下是1-可分辨的。可是集体$\{S\}$不包括所有分开i和j的线性无关有向截割组。而且也很清楚，E中那些边也不覆盖所有分开i和j的线性无关有向截割组。

当边组E是一个有向截割组，对于1-分辨率的一个条件由下一定理所示时就变得很简单了。

定理14—16：设G是从i到j的一个单输入单输出连通图。假使一个边组E是G中分开i和j的一个有向截割组。当且仅当G的每一条边是在$\{S\}$中的一个截割组，而$\{S\}$是分开i和j的由E中那些边所覆盖的所有有向截割组的一个集体时，G在E下是1-可分辨的。

证，假使G在E下是1-可分辨的，那么定理14—7说G中每条边是在$\{S\}$中的一个截割组中时，G在E下是1-可分辨的。

假使G在E下不是1-可分辨的，那么至少有一个含有一个点以上的点组V_I，其中的那些点不能由$\{S\}$中的任何一个截割组所分开。另一方面，每条边至少在$\{S\}$中的一个有向截割组中。所以在V_I中那些点之间不会有相连接的边。为了要有这样的一个组V_I，在$\{S\}$中一定有两个有向截割组$S_1=E(V_1\times\overline{V}_1)$和$S_2=E(V_2\times\overline{V}_2)$，而且是

$$i\in V_1,\quad j\in\overline{V}_1$$

$$i\in V_2,\quad j\in\overline{V}_2$$

和

$$V_I=\overline{V}_1\cap V_2$$

如图14—15所示。注意V_1，V_I和V_2是些点不共用组，它们包括G中所有点。

注意S_2中有一条边e在$E(V_I\times\overline{V}_2)$中任何一条边$e'$是在$E$之中。就容易看出有一个由边$e'$所覆盖的有向截割组，它分开$V_I$。所以在$E(V_I\times\overline{V}_2)$中没有一条边是属于有向截割组

E的。可是根据假定，E是一个分开 i 和 j 的有向截割组。我们知道S中的边e是在有向截割组E中，它位于V_1的一个点与$\overline{V}_2$中的一个点之间，并且我们也知道E不能分开V_I中的那些点。而且我们可以看出因为S_1和S_2是由E中那些边所覆盖的有向截割组，E不能分开$\overline{V}_2$和V_1中的任何点，因此E一定在S_1和S_2之中。这就意味着E包含着$E(V_1\times\overline{V}_2)$或$E(V_1\times V_I)$，这是一个矛盾。所以在$G$中没有这样的组$V_I$。这就证明了这个定理。▎

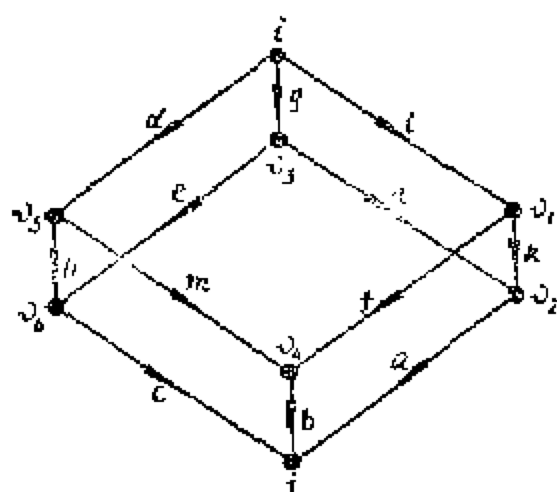


图 14—14

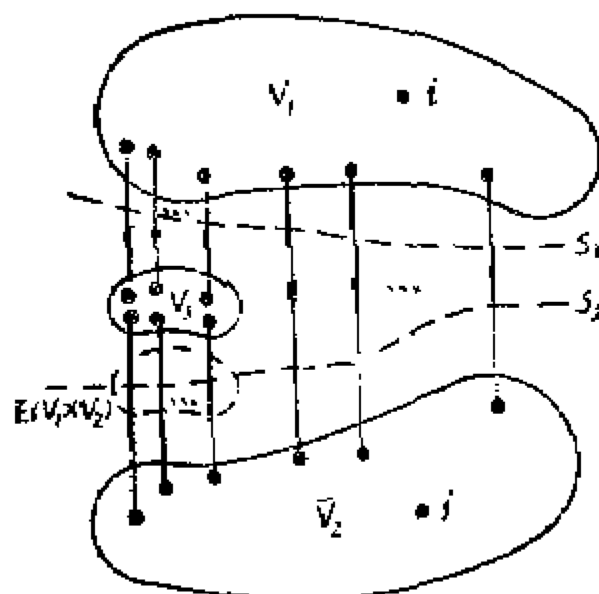


图 14—15

注意在前面的例子中，E是分开 i 和 j 的一个有向截割组。再每条边是在集体$\{S\}$中一个截割组中。因此根据定理14—16，单输入单输出连通图在E下是1-可分辨的。

现在让我们再研究些对找出故障点的实际测试直接有关的可测组。设$\{V\}$是对于组E中所有边指定测试点所产生的所有可能的可测组的一个集体。我们已经用由$\{V\}$得到的一个D-划分中一些组来规定一个单输入单输出连通图的1-分辨率。当我们研究$\{V\}$中的可测组时，我们会看到在$\{V\}$中有些真子集体$\{V\}$，由$\{V\}$所得到的一个D-划分是与由$\{V\}$所得到的一样。例如，设想图 14—16 中单输入单输出连通图G。设$E=(a,\ b)$，那么$\{V\}$包括$V(i\times a^+)=(i)$，$V(a^-\times j)=(1,2,j)$，$V(i\times b^+)=(i,2)$，$V(b^-\times j)=(j)$，$V(a^-\times b^+)=(2)$。由上面这些可测组所得到的D-划分中的非零组为：

$$V(i\times a^+)\cap\overline{V(a^-\times j)}\cap V(i\times b^+)\cap\overline{V(b^-\times j)}\cap\overline{V(a^-\times b^+)}=(i)$$

$$\overline{V(i\times a^+)}\cap V(a^-\times j)\cap V(i\times b^+)\cap\overline{V(b^-\times j)}\cap V(a^-\times b^+)=(2)$$

$$\overline{V(i\times a^+)}\cap V(a^-\times j)\cap\overline{V(i\times b^+)}\cap V(b^-\times j)\cap\overline{V(a^-\times b^+)}=(j)$$

$$\overline{V(i\times a^+)}\cap V(a^-\times j)\cap\overline{V(i\times b^+)}\cap\overline{V(b^-\times j)}\cap\overline{V(a^-\times b^+)}=(1)$$

只用$V(i\times a^+)$，$V(i\times b^+)$和$V(b^-\times j)$可以得到相同的D-划分。这就意味着靠了对应着$V(i\times a^+)$，$V(i\times b^+)$和$V(b^-\times j)$的 3 次测试，而不是对应着$\{V\}$中的所有非零可测组进行的 5 次测试，我们可以得到为用诊断用的所有必要信息。为了研究诊断所需的必要测试次数，我们对$\{V\}$的生成元下定义如下：

设$\{V\}$是组E所产生的所有可测组的一个集体。又设$\{V'\}$是$\{V\}$的一个子集体。如果（1）由$\{V'\}$所得到的一个D-划分与由$\{V\}$所得到的相同，和（2）由$\{V'\}$的任何子集体所得到一个D-划分与由$\{V\}$所得到的不相同，那么$\{V'\}$叫作$\{V\}$的生成元的一个集体。

在前面的例子中，由图14—16中单输入单输出连通图G的组$E=(a,b)$所产生的非零可测组的一个集体$\{V\}$是

$\{V\}=\{V(i\times a^+),\ V(a^-\times j),\ V(i\times b^+),\ V(b^-\times j),\ V(a^-\times b^+)\}$。由$\{V\}$所组

成的一个D-划分$\{D\}$为

$$\{D\}=\{(i),(2),(j),(1)\}$$

$\{V\}$的一个子集体$\{V(i\times a^+),V(i\times b^+),V(b^-\times j)\}$也给出相同的$D$-划分。可是$\{V(i\times a^+),V(i\times b^+),V(b^-\times j)\}$的任何一个真子集体不能给出相同的$D$-划分。所以$\{V(i\times a^+),V(i\times b^-),V(b^-\times j)\}$是$\{V\}$的生成元的一个集体。

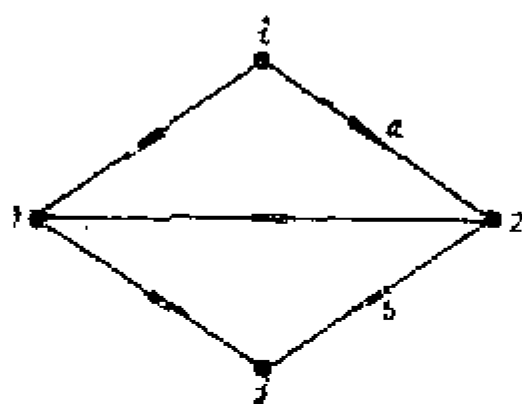


图 14—15

对于一个给出的$\{V\}$，可能有许多生成元集体。而且，在每一个$\{V\}$的生成元集体的组数可能是不同的。因为每一个可测组对应着诊断用的一次测试，并且一个$\{V\}$的生成元集体给出一个所需要的D-划分，所以诊断所需的测试次数是与$\{V\}$中的生成元的一个集体中的组数相同的。

有了生成元的定义后，我们可以有一个新定理。

定理14—17：对于一个从i到j的单输入单输出连通图中的一个组$E=(e_1,e_2,\cdots,e_k)$而言，可测组$V(i\times e_p^+)$和$V(e_p^-\times j)$所组成的一个集体（$p=1,2,\cdots,k$）含有$\{V\}$中生成元的一个集体。

证：因为$\{V\}$中的一个可测组是$V(i\times e_p^+)$，$V(e_p^-\times j)$，和$V(e_p^-\times e_q^+)$之中的一个、这就需要证明用不着$V(e_p^-\times e_q^+)$来得到$\{V\}$中生成元的一个集体。根据可测组的定义，$V(e_p^-\times e_q^+)$是一个点组，它至少含在一个$M(e_p^-\times e_q^+)$型的连通有向M图中。另一方面，$V(i\times e_q^+)$包括那些点，它们至少含在一个$M(i\times e_q^+)$型的连通有向M图中。而且我们知道，在一个从i到j的单输入单输出连通图中有一条从i到$v(e_p^-)$的有向路径。因此对于$M(e_p^-\times e_q^+)$型的连通有向M图中任何一个点v就有一个$M(i\times e_q^+)$型的连通有向图，它含有v。所以

$$V(e_p^-\times e_q^+)\subset V(i\times e_q^+) \tag{14—7}$$

同样

$$V(e_p^-\times e_q^+)\subset V(e_p^-\times j) \tag{14—8}$$

因此

$$V(e_p^-\times e_q^+)\subset V(i\times e_q^+)\cap V(e_p^-\times j) \tag{14—9}$$

设v'是$V(i\times e_q^+)\cap V(e_p^-\times j)$中的一个点，那么就有一个$M(e_p^-\times v')$型的连通有向$M$图和一个$M(v'\times e_q^+)$型连通有向$M$图。因此有一个$M(e_p^-\times e_q^+)$型的连通有向$M$图，它含有$v'$。所以由式（14—9）得

$$V(e_p^-\times e_q^+)=V(i\times e_q^+)\cap V(e_p^-\times j) \tag{14—10}$$

因此如果集体含有$V(i\times e_q^+)$和$V(e_p^-\times j)$的话，在集体中就不需要有$V(e_p^-\times e_q^+)$来包含$\{V\}$的生成元的一个集体。同样，我们可以证明在$\{V\}$的生成元集体中不需要$V(e_q^-\times e_q^+)$，$V(e_p^+\times e_q^-)$和$V(e_p^-\times e_q^+)$。▌

定理14—18：如果组E中一条边e覆盖一个有向截割组，那么$V(i\times e^+)$和$V(e^-\times j)$两者中只有一组可以在$\{V\}$的生成元一个集体中。

证：因为边e只覆盖一个有向截割组，$V(i\times e^+)\cup V(e^-\times j)$是所有点的点组。因此

$$V(i\times e^+)=\overline{V(e^-\times j)} \tag{14—11}$$

这就表明这两个可测组中只有一组是组成一个D-划分所需要的。▌

例如，在例14—4中图14—6中的单输入单输出连通图中，如果$E=(b,h)$，那么边b只覆盖一个有向截割组$S=(a,b)$。这可以看出

$$V(1\times b^+)=(1),\quad V(b^-\times 6)=(2,3,4,5,6)$$

这就表明$V(1\times b^+)=\overline{V(b^-\times 6)}$。另一方面，边$h$覆盖两个有向截割组$S_1(h)$和$S_3(h)$，与边$h$有联系的可测组为

$$V(1\times h^{+})=(1,\ 2,\ 4),\ V(h^{-}\times 6)=(6)$$

我们可以清楚地看到$\{V\}$的生成元的一个集体是$\{V(1\times b^{+}),V(1\times h^{+}),V(h^{-}\times 6)\}$。从这个定理，我们可以有下一定理。

定理14—19：对于一个组$E=(e_1,e_2,\cdots,e_k)$，在$\{V\}$的生成元的一个集体中组数最多为$2k$，其中k是组E中的边数。

下面我们通过一个铁路车站电气集中联锁的例子说明正确进行故障诊断的过程。首先对一个系统达到分辨率为1的子系统诊断，然后再对子系统进行诊断，直至进行到找出故障点。

下面是一个道岔表示电路图，如图14—17，这可以看作是一个小系统。

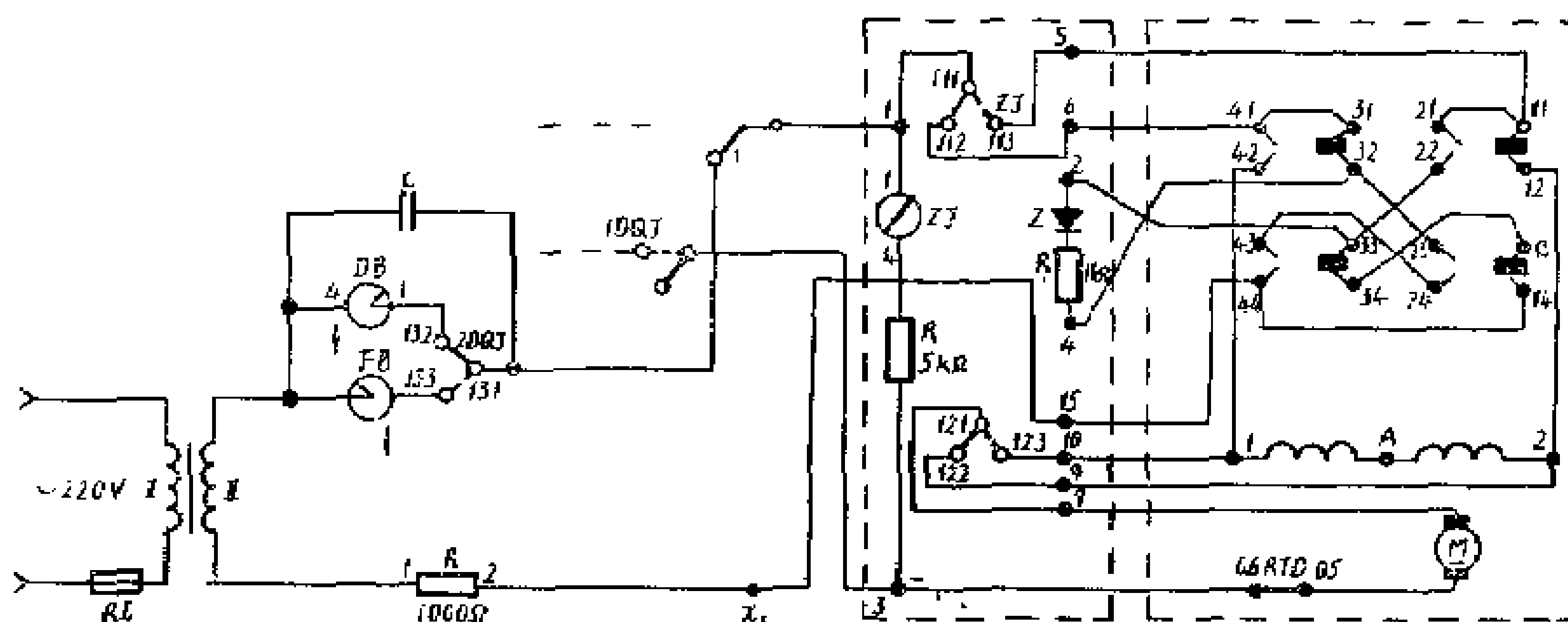


图 14—17

图14—18是道岔表示电路的等效原理图。这里要提起注意的是：故障诊断要求电路的正确连接，必须使第二道岔启动继电器$2DQJ$和转极继电器的连接点与道岔位置一致。

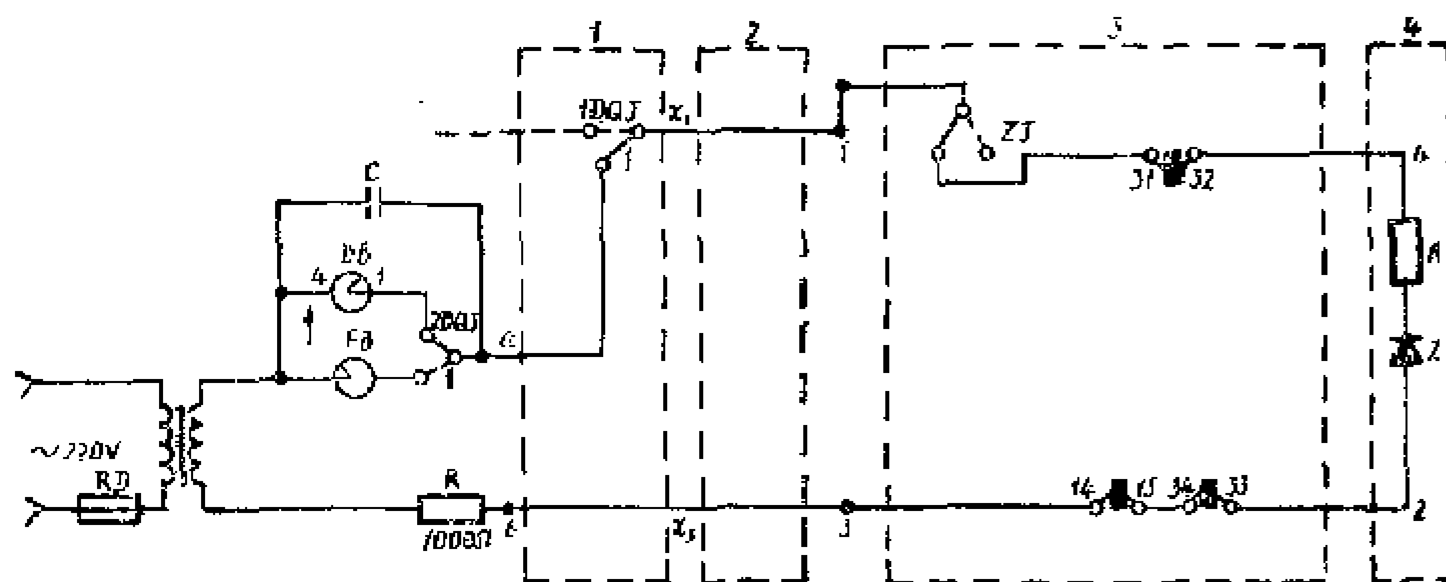


图 14—18

把该电路分块，使每个方块完成一种作用。这个电路是以交流220V为电源。在图中的状态，由定位表示继电器DBJ的吸起表示电路正常，DBJ落下表示电路故障。道岔表示电路经常在a，b两点右端出现故障。我们把在a，b测到的信号作为输入信号，而输出信号则由整流匣（图14—18中虚线框4）的两端取出。按其功能与要求作出诊断分辨到某一方块的方块图如图14—19。

把图14—19转化为图论中的有向图，如图14—20所示。

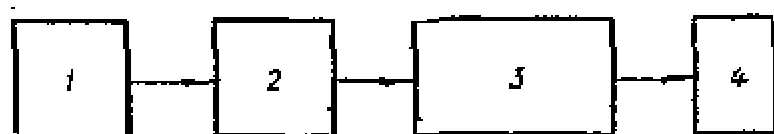


图 14—19

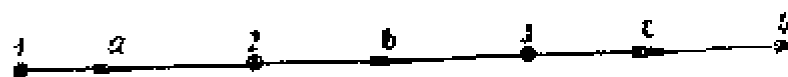


图 14—20

要使图14—20中的有向图有1-分辨率，对这个没有有向回路且含4个点的有向图，只要分开点1和点4的3个有向截割组由指定边组E覆盖就行了。显然$E=(a, b, c)$。由于组E中每条边都只覆盖一个有向截割组，所以3个可测组即可组成D-划分，使之达到1-分辨率。对子系统的诊断可采用故障诊断的其它方法。这里就不一一赘述了。

第四节 测 试 门

在上节中，我们观察到当一个单输入单输出连通图有一个有向回路时，就不可能得到1-分辨率。所以为了对这样一个图得到1-分辨率，就必须用测试点以外的其它方法，这就是测试门的作用。下面是测试门的定义。

指定给一条边e的测试门的意思是，当需要时，边e可以随时删除。

例如，如果我们在图14—21的单输入单输出连通图G中对边e指定一个测试门时，那么可测组$V(i\times j)=(i, j, 1, 2)$可以用测试门来改变成$(i, 2, j)$。这就是，由于测试门，我们可以改变$G$成为如图14—22所示的那样一个图。

为了表明测试门对可测组的影响，我们用一个下标E，如$V(i\times j)_E$表示指定给组E中那些边的测试门正在起作用。

符号$V(i\times j)_E$是在组E中删除所有边后得到的一个子图的一个可测组。

当在一个组E的所有边上都指定有测试门时，我们可以删除E中某些边而产生不同的可测组。

例14—8 图14—23中有一个单输入单输出连通图。假使$E=(e_1, e_2, e_3, e_7)$，那么我们有下列可测组：

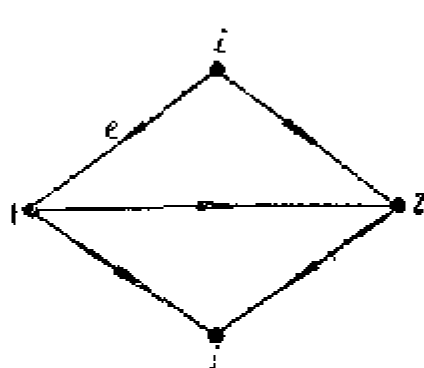


图 14—21

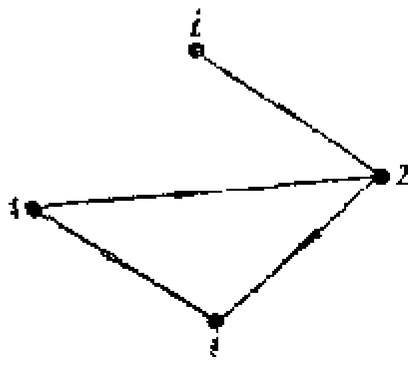


图 14—22

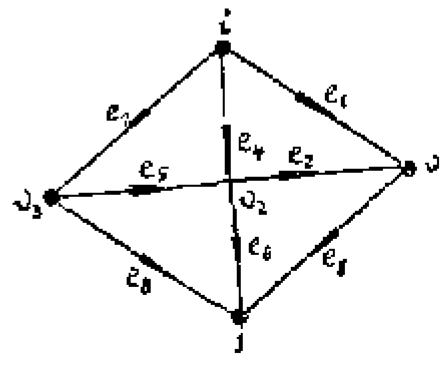


图 14—23

$$V(i\times j)e_1e_2e_3=V(i\times j)e_1e_2=V(i\times j)e_3=(i, v_2, v_3, j)V(i\times j)e_7$$
$$=(i, v_1, v_2, j)$$
$$V(i\times j)e_3, e_7=V(i\times j)e_1e_2e_7=V(i\times j)e_1e_2e_3e_7=(i, v_2, j)$$

为了研究测试门的一些特性，我们设想一个只用测试门的单输入单输出连通图。为了方

便起见，我们用下面的定义。

所谓门组E的可测组意思是由于在组E的所有边上指定有测试门而产生的所有可能的可测组。

在例14—8中，可测组$V(i\times j)e_3$，$V(i\times j)e_7$和$V(i\times j)e_3e_7$，是由门组(e_1,e_2,e_3,e_7)所产生的可测组。

下面的定理是从所有可能选择一些边的测试门不起作用的观点出发的。

定理14—20：由门组E的各别的可测组组数最多为2^k-1，其中k是组E中的边数。

设想一个从i到j的单输入单输出连通图中的可测组$V(i\times j)_E$。假使对于E的每个真子组E'，$V(i\times j)_E$与$V(i\times j)_{E'}$不同，那么我们说E是$V(i\times j)_E$的最小组。例如，在例14—3中考虑$E=(e_1,e_2,e_3,e_7)$，我们有$V(i\times j)_{E'}=V(i\times j)_E$。所以$E$不是关于$V(i\times j)_E$的最小组，而$E'$是关于$V(i\times j)_{E'}$的最小组。

在例14—8中

$V(i\times j)e_3e_7=V(i\times j)e_1e_2e_7=(i,v_2,j)$

其中(e_3,e_7)和(e_1,e_2,e_7)都是关于(i,v_2,j)的最小组。这个例子表明对于一个可测组的最小组可能不是唯一的。可是最小组有一个有趣的特性，由下一定理给出。

定理14—21：设E是一个从i到j的单输入单输出连通图G中关于$V(i\times j)_E(\neq\phi)$的最小组，那么从G中删除E中所有边后，结果的图还是连通的。

证：设g是从G中删除E中所有边后所得到的图，假设g是分离的。设g_1和g_2是g的两个子图，而$V(g_1)=V(i\times j)_E$，并且g_1和g_2不是连通的。注意G的$V(i\times j)_E$等于g的$V(i\times j)$。设v是g_2中一个点，因为G是一个从i到j的单输入单输出连通图，至少有一个$M(i\times j)$型的M图含有g_2中的点v。所以在M图中至少有两条边e_1和e_2，它们既不在g_1之中，又不在g_2之中，如图14—24所示。我们可看出，把e_1插入到g后，结果图的$V(i\times j)$是与g的$V(i\times j)$相同的。这就是说$V(i\times j)_E$等于G中的$V(i\times j)_{E'}$，其中$E'=E\ominus(e_1)$，所以E不是关于$V(i\times j)_E$的最小组。这是与假定所说E是关于$V(i\times j)_E$的最小组相矛盾的。因此这个定理是正确的。▌

由门组E的可测组的一个集体中，我们可以象前一节那样得到一个D-划分。例如，由一个门组$E=(e_1,e_2,e_3,e_7)$的可测组$V(i\times j)e_3$，$V(i\times j)e_7$和$V(i\times j)e_3e_7$的一个集体，我们可以得到一个D-划分如下：

$$V(i\times j)e_3\cap V(i\times j)e_7\cap V(i\times j)e_3e_7=(i,v_2,j)$$
$$V(i\times j)e_3\cap V(i\times j)e_7\cap\overline{V(i\times j)e_3e_7}=\phi$$
$$V(i\times j)e_3\cap\overline{V(i\times j)e_7}\cap V(i\times j)e_3e_7=\phi$$
$$V(i\times j)e_3\cap\overline{V(i\times j)e_7}\cap\overline{V(i\times j)e_3e_7}=(v_3)$$
$$\overline{V(i\times j)e_3}\cap V(i\times j)e_7\cap V(i\times j)e_3e_7=\phi$$
$$\overline{V(i\times j)e_3}\cap V(i\times j)e_7\cap\overline{V(i\times j)e_3e_7}=(v_1)$$
$$\overline{V(i\times j)e_3}\cap\overline{V(i\times j)e_7}\cap V(i\times j)e_3e_7=\phi$$
$$\overline{V(i\times j)e_3}\cap\overline{V(i\times j)e_7}\cap\overline{V(i\times j)e_3e_7}=\phi$$

注意一个可测组$V(i\times j)_E$的特性与前一节的相同。就是当在点i注入一信号时，由于指定给E中那些边的所有测试门发生作用，在点j所监视到的信号将只通过$V(i\times j)_E$中那些点。因此在前一节中关于可测组所有特性可适用于$V(i\times j)_E$中的可测组。例如，如果在从i到j的一个单输入单输出连通图G中有一对点v_r和v_s，而且每一个可测组(在一个

门组E的所有可测组的一个集体中）含有v_r和v_s中的一点的将含有其它一点的话，那么G不是1-可分辨的（定理14—7）。因为在一个门组E的可测组的一个集体中的每个可测组$V(i\times j)_E$都含有点i和j，一个从i到j的单输入单输出连通图在任何一个门组E下不是1-可分辨的。如果我们忽略点i和j中的一点，在一个门组E下，有些单输入单输出连通图显然是1-可分辨的。为了研究这样的单输入单输出连通图的门组，我们需要另一个定义如下。

如果不管点j，一个从i到j的单输入单输出连通图G在一个门组E下是1-(i,j)-可分辨的。因此G在组E下是1-可分辨的。

例14—9　在图14—25中从点1到4的单输入单输出连通图中，取一个门组$E=(b,d)$，E的可测组为：

$$V(1\times 4)_b=(1,2,4)$$
$$V(1\times 4)_d=(1,3,4)$$
$$V(1\times 4)_{bd}=\phi$$

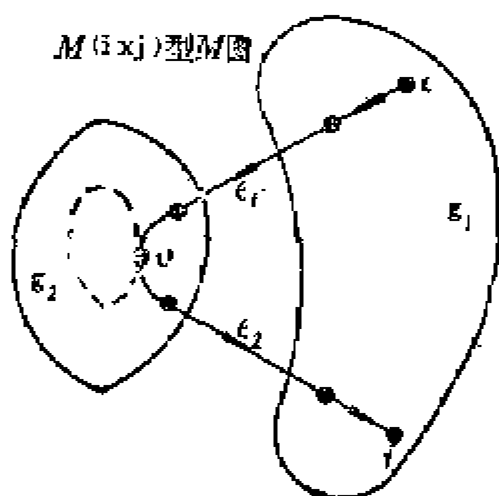


图 14—24

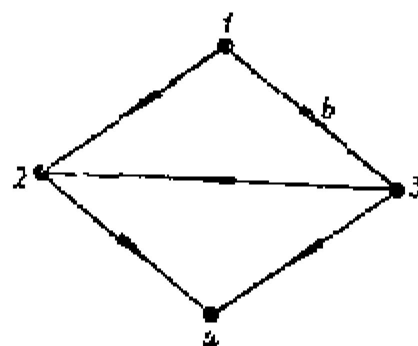


图 14—25

这些可测组的一个D-划分包括：

$$V(1\times4)_b\cap V(1\times4)_d=(1,\ 4)$$
$$V(1\times4)_b\cap\overline{V(1\times4)_d}=(2)$$
$$\overline{V(1\times4)_b}\cap V(1\times4)_d=(3)$$
$$\overline{V(1\times4)_b}\cap\overline{V(1\times4)_d}=\phi$$

如果我们不管点4，在上面D-划分中的每一个组最多含有一个点。所以如果我们忽略点4，G在E下是1-可分辨的。因此G在E下是1—(1，4)-可分辨的。

用了上面这个定义，我们可以修改定理14—7如下，这样它可以适用于测试门的情况。

定理14—22：设G是一个从i到j的单输入单输出连通图，和E是一个边组。又设$\{V\}$是给E中所有边指定着测试门后所产生的所有可测组的一个集体。那么当且仅当对于G中每一对点v_r和v_s没有考虑点j（就是$v_r\neq j$和$v_s\neq j$）和有一个可测组$V\in\{V\}$，而V含有v_r和v_s，但不含有两者时，G在E下是1-(i,j)-可分辨的。

这个定理的证明差不多与定理14—7的证明一样。

在例14—9中，图14—25中单输入单输出连通图的组$E=(b,d)$所产生的可测组的一个集体$\{V\}$是

$$\{V\}=\{V(1\times4)_b,\ V(1\times4)_d\}=\{(1,\ 2,\ 4),\ (1,\ 3,\ 4)\}$$

所以对于任何一对点，我们选择除点4以外的点，在$\{V\}$中就有一个可测组，它只含有这两点中的一个点。因此根据定理14—22，G在$(b,\ d)$下是1-$(i\ j)$-可分辨的。

过去已经讲过，一个分开 i 和 j 的半截割可以表达为$E(V_i \times \overline{V}_i)$，其中 $i \in V_i$，$j \in \overline{V}_i$。现在如果 $E(\overline{V}_i \times V_i)=\phi$，那么半截割$E(V_i \times \overline{V}_i)$变成一个分开 i 和 j 的有向截割组。由下一定理我们会看到这样一个半截割对于1-(i，j)-分辨率是重要的。

定理14—23：设G是一个从 i 到 j 的单输入单输出连通图在 i 和 j 之间没有直接相连的边。如果一个门组E不含有一个分开 i 和 j 的半截割作为一个子组，那么G 在门组 E 下不是1-(i，j)-可分辨的。

证：因为E不含有一个分开 i 和 j 的半截割作为一个子组，从 i 到 j 有一条有向路径P，它不含有E中的边。根据假定，G没有直接连接 i 和 j 之间的边，所以路径P一定至少含有除 i 和 j 以外的另一个点 v。因此由 E所得到的每个可测组含有点 v 连同 i 和 j。根据定理14—7，即使不管点 j，G也不是1-可分辨的。因此G在门组E下不是1-(i，j)-可分辨的。■

例14—10　图14—26中有一个从 i 到 j 的单输入单输出连通图。设一个门组E是（a，d，f）。因为（a，d，f）不含有分开 i 和 j 的半截割，根据定理14—23，G在（a，d，f）下不是1-(i，j)-可分辨的。现在我们进行检查。

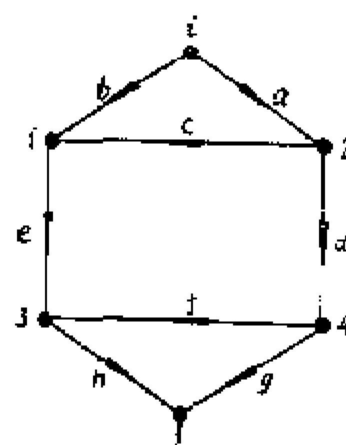


图 14—26

由门组（a，d，f）得到的可测组为

$V(i\times j)_a=V(i\times j)_{ad}=V(i\times j)_{af}=V(i\times j)_{adf}=(i,1,3,j)$

$V(i\times j)_f=(i,1,2,3,4,j)$

$V(i\times j)_d=V(i\times j)_{df}=(i,1,2,3,j)$

这些可测组的一个D-划分中含有一个组D：

$$D=V(i\times j)_a \cap V(i\times j)_d \cap V(i\times j)_{af}=(i,1,3,j)$$

因此G在门组（a，d，f）下不是1-(i，j)-可分辨的。

根据定理14—23，为了一个单输入单输出连通图在E下是1-(i，j)-可分辨起见，一个组E必须含有一个从 i 到 j 的半截割。所以要研究能给出1-(i，j)-分辨率的组的一个边组E，我们假定E本身是一个有向截割组 $S=E(V_i \times \overline{V}_i)$，其中$i \in V_i$，$j \in \overline{V}_i$。设$E=(e_1, e_2, \cdots, e_k)$。如果我们从单输入单输出连通图$G$中删去 $e_p(1\leqslant p\leqslant k)$ 以外的其它所有边，那么一个在点 i 处注入、在点 j 处监视的测试信号一定通过边 e_p。这就导致下一定理。

定理14—24：设G是一个从 i 到 j 的单输入单输出连通图。设$E=(e_1, e_2, \cdots, e_k)$是一个有向截割组$E(V_i \times \overline{V}_i)$，其中$i \in V_i$，$j \in \overline{V}_i$，那么对于$p=1, 2, \cdots, k$，

$$V(i\times j)_{(\overline{e}_p)}=V(i\times e_p^+)\cup V(e_p^-\times j) \qquad (14—12)$$

式中 $(\overline{e}_p)=E\ominus(e_p)$。设$E_q$是$E$的一个子组，那么

$$V(i\times j)_{\overline{E}_q}=\bigcup_{e_p \in E_q} V(i\times j)_{(\overline{e}_p)} \qquad (14—13)$$

式中 $\overline{E}_q=E\ominus E_q$。

证：我们知道$V(i\times e_p^+)$是一个点组，其中每一个点至少是在$M(i\times e_p^+)$型的一个连通有向M图中，和$V(e_p^-\times j)$是另一个点组，其中每一个点至少是在 $M(e_p^-\times j)$型的一个连通有向M图中（参看第十四章第三节）这个定理的第一部分，从前面的讨论中可知是明显的，不必

再证明。对于第二部分，我们注意到一个从 i 到 j 的有向路径通过 $V(i\times j)_{\bar{E}_q}$ 中的那些点，它一定只含有E_q中的一条边。因此对于某些 $e_t\in E_q$，在 $V(i\times j)_{\bar{E}_q}$中的每一个点是在 $V(i\times j)_{(\bar{e}_p)}$之中。因此在$V(i\times j)_{(\bar{e}_q)}$中的任何一个点，也很清楚一定是在$V(i\times j)_{\bar{E}_q}$之中。▮

定理14—24不仅给出了$V(i\times j)_{(\bar{e})}$ 型可测组与第十四章第三节中的可测组之间的关系，而且也表明为了求取在一个给出组E下的一个D-划分，我们只需知道$V(i\times j)_{(\bar{e})}$ 型可测组就够了。

例14—11 图14—27是一个从 i 到 j 的单输入单输出连通图。

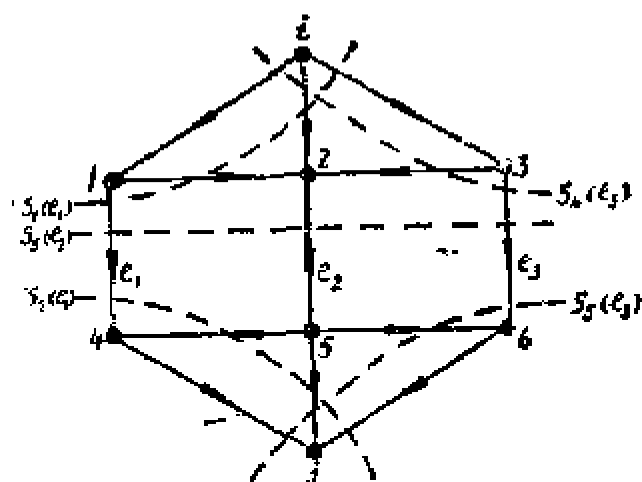


图 14—27

设$E=(e_1,\ e_2,\ e_3)$。我们可以知道：

$V(i\times e_1^+)=(i,\ 1)$

$V(e_1^-\times j)=(4,\ j)$

$V(i\times e_2^+)=(i,\ 1,\ 2,\ 3)$

$V(e_2^-\times j)=(4,\ 5,\ 6,\ j)$

$V(i\times e_3^+)=(i,\ 3)$

$V(e_3^-\times j)=(6,\ j)$

根据定理14—24，我们可以得到

$$V(i\times j)(e_2e_3)=V(i\times j)(\overline{e_1})=V(i\times e_1^+)\cup V(e_1^-\times j)=(i,\ 1,\ 4,\ j)$$

$$V(i\times j)(e_1e_3)=V(i\times j)(\overline{e_2})=V(i\times e_2^+)\cup V(e_2^-\times j)=(i,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ j)$$

$$V(i\times j)(e_1e_2)=V(i\times j)(\overline{e_3})=V(i\times e_3^+)\cup V(e_3^-\times j)=(i,\ 3,\ 6,\ j)$$

$$V(i\times j)(e_1)=V(i\times j)(\overline{e_2e_3})=V(i\times j)(\overline{e_2})\cup V(i\times j)(\overline{e_3})=(i,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ j)$$

$$V(i\times j)(e_2)=V(i\times j)(\overline{e_1e_3})=V(i\times j)(\overline{e_1})\cup V(i\times j)(\overline{e_3})=(i,\ 1,\ 3,\ 4,\ 6,\ j)$$

$$V(i\times j)(e_3)=V(i\times j)(\overline{e_1e_2})=V(i\times j)(\overline{e_1})\cup V(i\times j)(\overline{e_2})=(i,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ j)$$

当我们应用测试点时，根据定理14—16知道如果边组E是分开 i 和 j 的一个有向截割组，和一个从 i 到 j 的单输入单输出连通图G中的每一条边至少在E覆盖下的$\{S\}$中的一个有向截割组时，这个G在E下是1-可分辨的。当我们在E的那些边上指定测试门时，情况就不同了。例如，在图14—27中的单输入单输出连通图中，由$E=(e_1,e_2,\ e_3)$覆盖的有向截割组的集体$\{S\}$包括截割组S_1，S_2，S_3，S_4和S_5，如图所示。注意E是一个分开 i 和 j 的有向截割组。容易看出单输入单输出连通图中每一条边是在$\{S\}$的一个截割组中。另一方面，考虑到点 1 和 4 的点组（1，4），我们可以看出对于E的每一个子组E'，或者（1，4）$\subset V(i\times j)E$，或者（1，4）$\subset\overline{V(i\times j)E'}$。所以根据定理14—22，$G$在（$e_1$，$e_2$，$e_3$）下不是1-（$i$，$j$）-可分辨的。这个例子表明当我们应用测试门时，需要图的每一条边是在$\{S\}$中的条件不足以说明一个单输入单输出连通图是1-（i，j）-可分辨的。但是，请注意下一定理的特性。

定理14—25：设G是一个从i到j的单输入单输出连通图，假使组E是一个有向截割组$E(V\times\overline{V})$，这样或者V只包括点i，或者$\overline{V}$只包括点j。那么，当且仅当G的每一条边是在$\{S\}$中的一个有向截割组中，其中$\{S\}$是E所覆盖的有向截割组的一个集体时，G是1-(i,j)-可分辨的。

证：根据定理14—24

$$V(i\times j)(\overline{e_p})=V(i\times e_p^+)\cup V(e_p^-\times j) \qquad (14-14)$$

其中$e_p\in E_c$ 假使$\overline{V}$只包括点j，那么$V(e_p^-\times j)=(j)$。因此对于所有边$e_p\in E$，$V(i\times j)(\overline{e_p})=V(i\times e_p^+)\cup(j)$。所以没有点$j$时，用测试门的$e_p$的可测组与用测试点的$e_p$的可测组相同。当$V$只包括点$i$时，$V(i\times j)(e_p)=(i)\cup V(e_p^-\times j)$。这表明当我们不管点$i$时，用测试门的$e_p$的可测组与用测试点的$e_p$的可测组相同。因此用测试点的在$E$下一个从$i$到$j$的单输入单输出连通图是1-可分辨的必要和充分条件对于这种情况也可以适用的，这就证明了这个定理。▮

例14—12 图14—28中有一个从i到j的不可分的单输入单输出连通图G，由$E=(e_1, e_2, e_3)$覆盖的有向截割组的集体$\{S\}$包括4个截割组S_1，S_2，S_3和S_4：

$$S_1=E\left\{(i,\ 1,\ 2,\ 4)\times\overline{(i,\ 1,\ 2,\ 4)}\right\}$$

$$S_2=E\left\{(i,\ 1,\ 3,\ 5)\times\overline{(i,\ 1,\ 3,\ 5)}\right\}$$

$$S_3=E\left\{(i,\ 2,\ 3,\ 6)\times\overline{(i,\ 2,\ 3,\ 6)}\right\}$$

$$S_4=E\left\{(\overline{j})\times(j)\right\}$$

如图14—28所示。

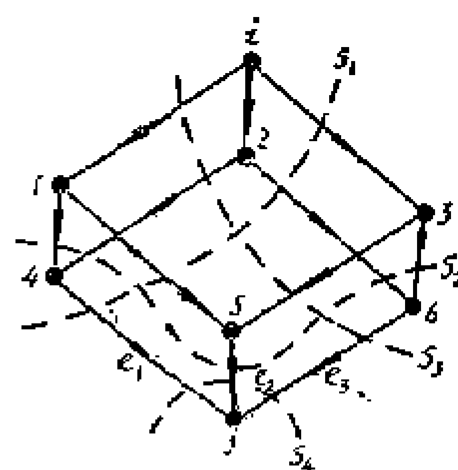


图 14—28

因为G中每条边是在$\{S\}$的一个截割组之中，所以根据定理14—26，G在(e_1, e_2, e_3)下是1-(i, j)-可分辨的。

定理14—26：设G是一个从i到j的单输入单输出连通图。假使组$E=E(V\times\overline{V})$是一个有向截割组。如果G的每一条边是在$\{S\}$的一个截割组中，那么G在E下或是1-(i, j)-可分辨的，或是2-(i, j)-可分辨的，其中$\{S\}$是E覆盖下的有向截割组的一个集体。

证：在G中把V中所有点合并起来得到一个单输入单输出连通图G'。那么很清楚，(S)的一个子集体是由E所覆盖的G'中的有向截割组的一个集体，以及G'中的每条边是在于集体的一个截割组中。因为$E=E(V'\times\overline{V}')$是$G'$中的一个有向截割组，其中$V'=(i)$和$\overline{V}'=\overline{V}$。根据定理14—25，$G'$在$E$下是1-$(i, j)$-可分辨的。同样，把$\overline{V}$中所有点合并起来得到一个新的单输入单输出连通图$G''$，它在$E$下是1-$(i, j)$-可分辨的。因此在$V$中的任何两点不能都在$E$下的一个$D$-划分中的一个组中。同样，$\overline{V}$中任何两个点也不能都在$E$下的一个$D$-划分中的一个组中。可是如果一个点是在$V$中，另一个点是在$\overline{V}$中，那么就不能保证这两个点是在$E$下的一个$D$-划分中的不同组中。因此$G$在$E$下或是1-$(i, j)$-可分辨的，或是2-$(i, j)$-可分辨的。▮

在图14—27所示的从i到j的那个单输入单输出连通图中，如果$E=(e_1, e_2, e_3)$，那么$\{S\}=\{S_1, S_2, S_3, S_4, S_5\}$，它是$E$覆盖下的有向截割组的一个集体。这很清楚$G$中每条边是在$\{S\}$中的一个截割组中。因此根据定理14—26，$G$在$E$下或是1-$(i, j)$-可分辨的，或是2-$(i, j)$-可分辨的。注意在$E$下一个$D$-划分的那些组是$(i, j)$，$(1,$

4），（2，5）和（3，6）。

第五节　接点开关电路中迂回电路的诊断

在接点开关电路中，人们关心的问题是输入到输出之间的传输情况。从输入端到输出端的全部通路中，除了设计规定的一条或若干条通路外，若还存在有其它通路，则这些通路就称为迂回电路。迂回电路会使某些设备——例如铁路行车自动控制系统中电气集中联锁电路——造成严重的后果，因此最好在电路设计与研制时就能够发现并给予防护。在大型接点开关电路中全部迂回电路的发现是一个比较困难的问题。

一个接点开关网络中两点间的迂回电路在其对应的拓扑图中可用两点间的某些路径来说明。因此迂回电路可利用路径集体的性质再加上工程上的一些规定而得到。在第十章中我们已经给出过几种获得路径集体的方法。这一节再给出一种更实用的方法，它可以编制为计算机程序来找寻一个线性图中两点间的所有路径。

首先介绍获得一个图中两点间全部路径方法的几个概念。

后继点和前趋点：设点v_i和点v_j是图$G=(V、E)$的两个点，(v_i, v_j)为从点v_i到点v_j的一条有向边。如果$(v_i, v_j)\in E$，则点v_j称为点v_i的后继点。点v_i的所有后继点的组用$\Gamma^+(v_i)$来表示。相反，如果$(v_j, v_i)\in E$，则点v_j称作点v_i的前趋点。点v_i的所有前趋点的组用$\Gamma^-(v_i)$来表示。

可达性：设$G=(V, E)$是任意图，点v_i和点v_j是图中两个点。如果在图中存在一条从点v_i到点v_j的路径，则点v_j认为是从点v_i可达的。相反，如果图中存在一条从点v_j到点v_i的路径则点v_j认为是从点v_i相反可达的。从v_i可达的或相反可达的点组分别用$\overset{*}{\Gamma}{}^+(v_i)$和$\overset{*}{\Gamma}{}^-(v_i)$来表示。下面叙述获得一个图中两点间全部路径的方法。

对于一个给定的有向图$G=(V, E)$，假定我们希望获得从一个给定的点v_p到另一个给定的点v_q的全部路径（这儿设点v_p和v_q是不同的，并且v_p点是从v_q点可到达的）。我们用M代表所需要的路径集体，采用下述方法来得到。

设$S_p=\{v_{s1}, v_{s2}, \cdots, v_{sk}\}$是$G$中$v_p$的所有后继点的组，并且在$G$中删去$v_p$点后获得的子图$H$中也是可以从$v_q$点相反可达的。对于$i=1, 2, \cdots, k$，设$M_i$代表$M$的一个子路径集体，在这个子路径集体中，第一条边是$(v_p, v_{si})$于是能够把原来获得$M$的问题分解成获得$k$个子问题$M_1, M_2, \cdots, M_k$，它们合在一起正好组成$M$。

考虑这些子集体之一时，我们注意到：

（一）$v_{si}=v_q$在这种情况下，M_i仅含有一条路径(v_p, v_q)。

（二）$v_{si}\neq v_q$，在这种情况下，M_i是由边(v_p, v_{si})和从v_{si}到v_q的所有路径的集体M_i'组成的。

在第一种情况下，M_i的获得是很明显的。在第二种情况下，问题减小到了对M_i'路径集体如何得到的问题。这个问题与原来的问题相比已经成为一个比原来问题小的问题。M_i'能以同样的方法继续被分解。重复这个过程，我们就能够从图G中找到从v_p点到v_q点的所有路径。

在一般情况下，上面的叙述可以描绘成一棵树。在这棵树中，根代表最初的问题，而树中的其它点代表各个子问题，这些子问题不断分解直至得到所有答案。

例14—13　在图14—29中，让我们考虑获得从点2到点4的所有路径问题。

这个问题的连续分解过程直到最后的答案全部显示在图14—30的树中。

这棵树的根代表原先的问题，在根点的内部，我们以实线画出了原先图G的子图H，是由删去点2得到的；星号代表了H上属于点4的相反可达集体，而虚线代表由星号点所决定的来自点2的边。在图14—30中的根有两个即点1和点3。因而原先的问题能分解为二个子问题，即包括子图H中从点1到点4和从点3到点4的所有路径的获得问题。这两个子问题由级为1（见第三章第三节）的两个树点来描述。在这棵树中，已经分配给路径的边由粗线代表。树中各点的后继点描述了子问题的分解，而悬挂点描述了最后获得的子问题。

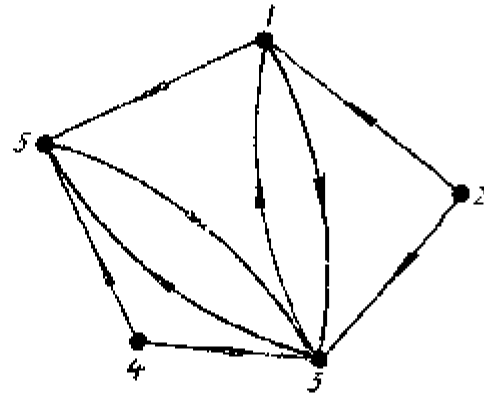


图 14—29

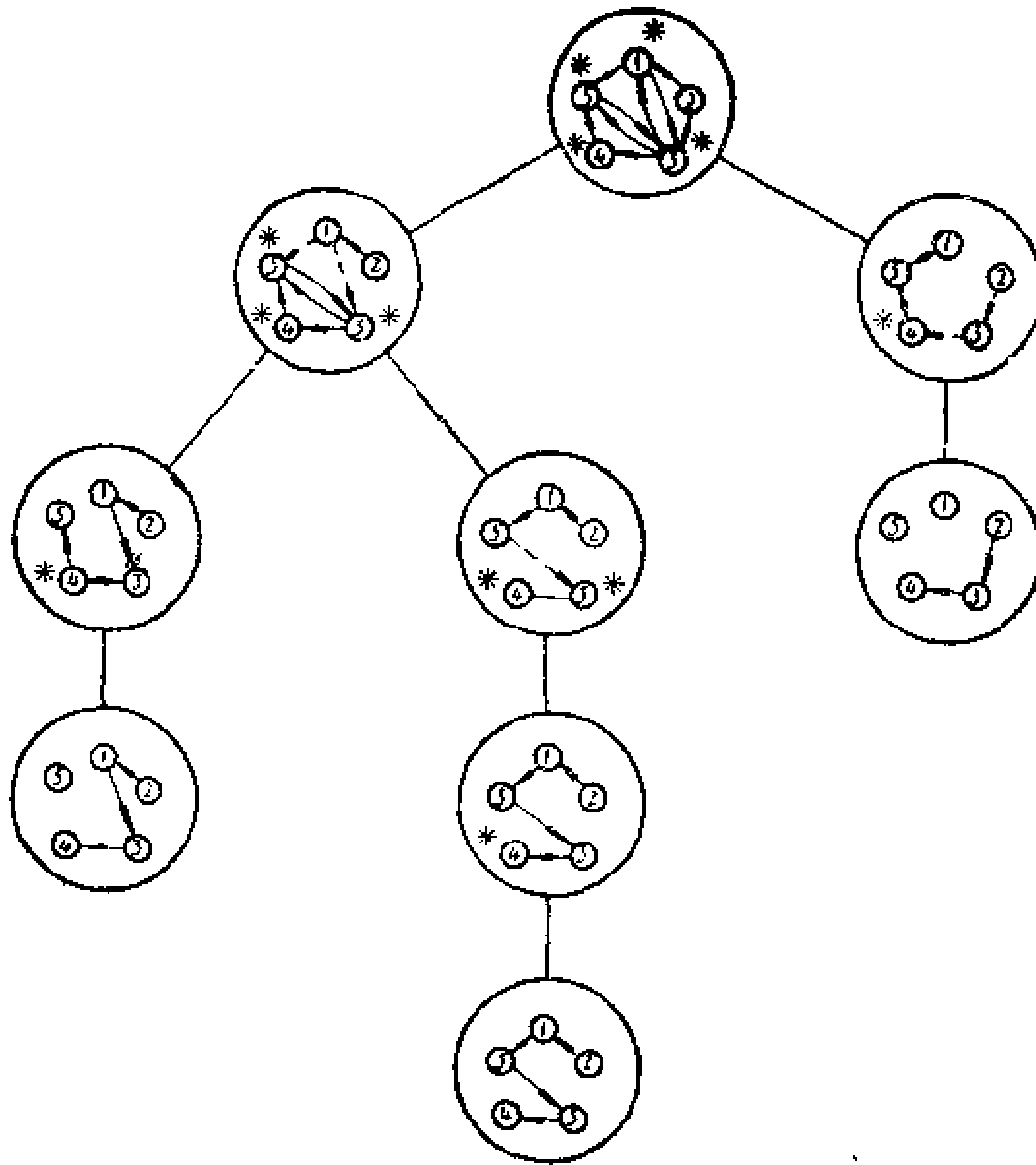


图 14—30

在大型网络中为了正确迅速地获得答案，必须有一个正确有效的方法搜索树中的各点。考虑到树中各点的搜索次序，会有两种搜索方式，其一是从根开始搜索所有级为1的点，然后再搜索所有级为2的点等等。这样的搜索过程就是在第十一章中所讲的广度第一搜索法。很明显，若给出的图有许多点和边，由此产生的树是相当大的，树中的点也会很多。这样一来广度第一搜索法就会影响到复杂的计算。

对一个给定的有限树进行系统的穿行，使得每条边正好穿过一次，每个点正好穿过一次的有效方法是图的深度第一搜索法。这种方法是当从根开始扫描各邻点后，尽快前进到尚未去过的邻点，而暂时不管邻接到根的其它点。对于前进到的点重复上述过程。换句话说，只

要可能，我们总是沿着图的一条路径通到一个新点，在返回之前我们尽可能前进到距离根最远的那个点。这种算法对于简化图论算法是很有用的。

这个穿越树的方法可以用图论的理论来说明。

设$G=(V, E)$是一棵有向树，那么定义G的横向次序是由下面规则决定的一种次序。

对于G中的两个点v_i和v_j

（一）如果$\Gamma^-(v_i)=\Gamma^-(v_j)$，则$v_i \propto v_j$或$v_j \propto x_i$；

（二）如果$\Gamma^-(v_i)\neq\Gamma^-(v_j)$，则$v_i$和$v_j$是不能够比较的。

这里，$\Gamma^-(v_i)$和$\Gamma^-(v_j)$代表点i和点j的前趋点$\propto$表示横向次序，因而，一棵树的横向次序提供了它的每一个点的后继点的总次序。

给出一个有向树$G=(V, E)$的横向次序$\propto$，我们可以构成同$\propto$一致的V的次序$\mathscr{R}$(符号$\mathscr{R}$表示一种关系，一种前后关系)。在$\mathscr{R}$中，对每一对（v_i，v_j）讲，因为$v_i \propto v_j$，同样有$v_i \mathscr{R} v_j$。这样，我们可由下面的规则决定一个总次序：

（一）如果树中v_i的级小于v_j的级，并且存在从v_i到v_j的路径，则$v_i \mathscr{R} v_j$；

（二）如果$v_i \propto v_j$，则对所有$v_i' \in \overset{*}{\Gamma}{}^+(v_i)$和所有$v_j' \in \overset{*}{\Gamma}{}^+(v_j)$，$v_i' \mathscr{R} x_j'$。

这里，$\overset{*}{\Gamma}{}^-(v_i)$和$\overset{*}{\Gamma}{}^+(v_j)$代表了树中从点$v_i$和点$v_j$可到达的点组。

这样，我们就可以把一个有限的有向树的所有点进行了有秩序的排列。

在图14—31中，点1和点4间的路径可由图14—32的树来描述并进行搜索。

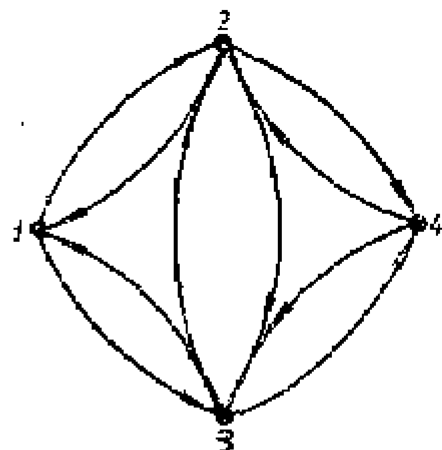
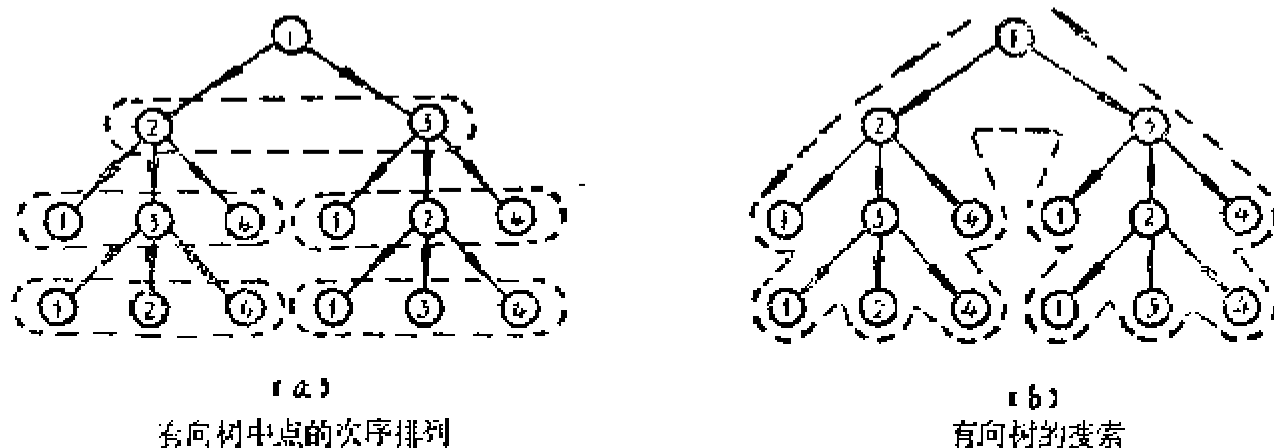


图 14—31

在图14—32（a）中，被$\propto$排列次序的点组是由虚线框限定的，而且当$v_i \propto v_j$时，点v_i被放到v_j的左方。于是按照上面的理论，这个有向树中的所有点就能以这样一种次序排列，即 1 2 1 3 1 2 4 4 3 1 2 1 3 4 4。这样的次序由深度第一搜索法可以完成，如图14—32（b）中所示。简单说明如下：

设v_s是正在被访问的点，如果v_s还存在没有被访问到的后继点，则前进到这些后继点中最小的（对横向次序$\propto$而言）；否则返回到v_s的前趋点。当返回到根时，这个过程就完成了。



（a）有向树中点的次序排列　　（b）有向树的搜索

图 14—32

由此我们可以看出深度第一搜索法有效地包含了上面有向树中的每一点。这对利用电子计算机搜索来讲是非常有用的方法。

在大型接点开关电路的研制与设计中有时会存在迂回电路，这需要及时诊断出来并在设计中采取措施防护，以防在使用中造成严重的后果。采用上述产生路径集体的理论和工程中的一些规定可以很容易地利用电子计算机找出电路中的迂回电路。

例14—14　在图14—33的电气集中联锁的选岔电路中，人工找寻按钮 a 和 c 点间的全部通路是非常困难而复杂的工作。但是利用图论中产生路径集体的算法和电子计算机就可以迅速地找出它的拓扑图（图14—34）中 a，c 两点间的94条路径。除了设计中规定的通路之外，其它的都是迂回电路，应当予以防护。

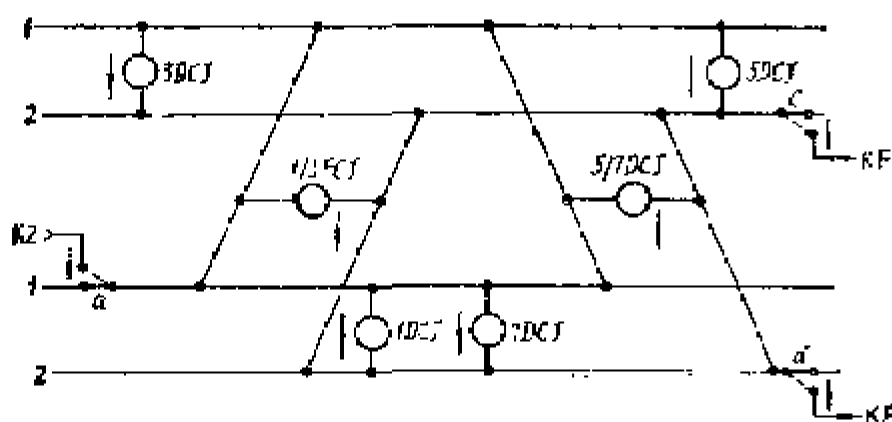


图　14—33

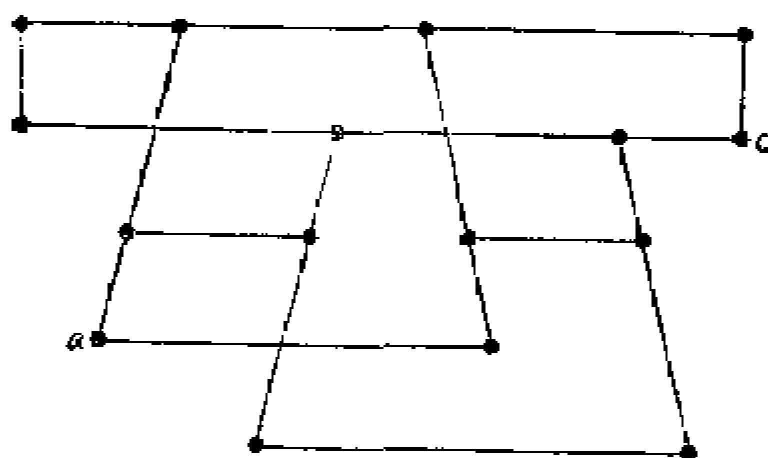


图　14—34

习　　题

14—1　当 $V_1=(1,\ 2,\ 3,\ 4)$，$V_2=(1,\ 3,\ 5)$

$V_3=(2,\ 3,\ 4)$，　$V_4=(1,\ 3,\ 4)$

$V_1\cup\overline{V}_1=(1,\ 2,\ 3,\ 4,\ 5)$

求 D-划分。

14—2　图14—35是一个从点 i 到点 j 的单输入单输出连通图。求组 $E=(a,\ b)$ 下所有可测组：

$V(i\times a^+)$，$V(a^-\times j)$，$V(i\times b^+)$，$V(b\ \times j)$，$V(a^+\times b^+)$，

$V(a^+\times b^-)$，$V(a^-\times b^+)$，$V(a^-\times b^-)$，$V(b^+\times a^+)$，$V(b^-\times a^-)$

$V(b^-\times a^-)$，$V(b^-\times a^-)$。

14—3　求题14—2中单输入单输出连通图中 $E=(a,\ b)$ 覆盖下的所有有向截割组。

14—4　在图14—36中从点 i 到点 j 的单输入单输出连通图中，求门组 $E=(a,\ b,\ c)$ 的所有可测组。

14—5　在图14—37中，从点 i 到点 j 的单输入单输出连通图中，给边 a 和 b 指定测试点，给边 c 和 d 指定测试门，求所有测试组。并问这个图在这种情况时是否是1-可分辨的？

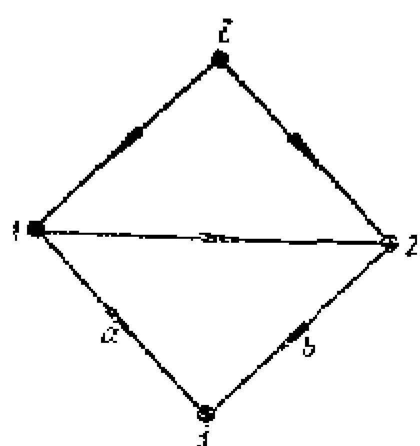


图 14—35

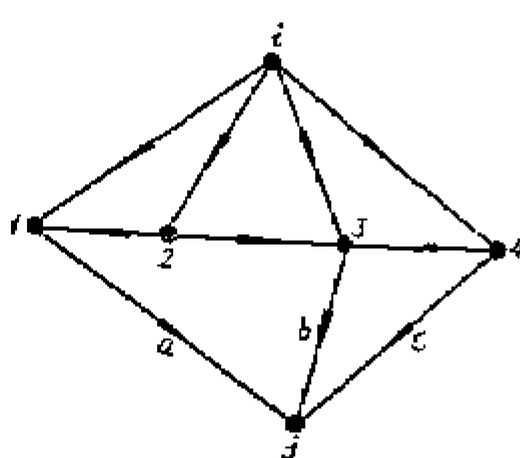


图 14—36

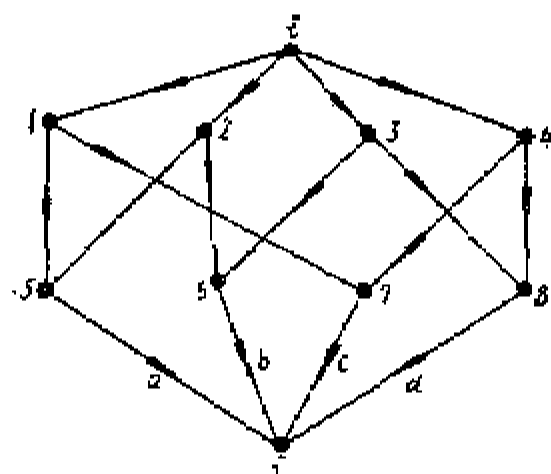


图 14—37

参 考 文 献

1. Narsingh Deo, Graph Theory With Applications to Engineering and Computer Science, Prentice-Hall, Inc., Englewood Cliffs, N.Y., 1974
2. Wataru Mayeda, Graph Theoty, Wiley-Interscience, New York, N.Y., 1972
3. Vinod Chachra, Prabhakar M.Chare, and James M.Moore, Applications of Graph Theory Algorithms, Elsevier North Holland, Inc., New York, N.Y., 1979
4. D.F.Robinson and L.R.Foulds, Digraphs: Theory and Techniques, Jordon and Breach Science Publishers, Inc., New York, N.Y., 1980
5. Haward Cravis, Communications Network Analysis, D.C.Heath and Co., Lexington, Mass., 1981
6. F.Harary, Graph Theory, Addison-wesley Publishing Co., Inc., Reading Mass., 1969
7. Lee M.Maxwell and Myril B.Reed, The Graph Theory: A Basis for Network Theory, Pergamon Press, New York, N.Y., 1971
8. Bernard Carre, Graphs and Networks, Clarendon Press, Oxford, 1979